N° d'ordre : 456 Année 1971

THÈSE

présentée

à l'Université Paul Sabatier de Toulouse

en vue de l'obtention

du Grade de Docteur ès Sciences Physiques

par

Jean-Claude IPPOLITO
Ingénieur E. N .S. E. E. H. T.
Docteur de Spécialité

CONTRIBUTION A L'IMPLANTATION DE FONCTIONS LOGIQUES SUR DES RÉSEAUX CELLULAIRES

Soutenue le 21 Décembre 1971 devant la Commission d'Examen

MM. J. LAGASSE Président
G. GRATELOUP
J. FLORINE Examinateurs
P. AZEMA

N° d'ordre : 456 Année 1971

THÈSE

présentée

à l'Université Paul Sabatier de Toulouse

en vue de l'obtention

du Grade de Docteur ès Sciences Physiques

par

Jean-Claude IPPOLITO

Ingénieur E. N .S. E. E. H. T.
Docteur de Spécialité

CONTRIBUTION A L'IMPLANTATION DE FONCTIONS LOGIQUES SUR DES RÉSEAUX CELLULAIRES

Soutenue le 21 Décembre 1971 devant la Commission d'Examen

MM. J. LAGASSE — Président
G. GRATELOUP, J. FLORINE, P. AZEMA — Examinateurs

Digitized by the Internet Archive
in 2026

https://archive.org/details/IA41543907_0034

UNIVERSITE PAUL SABATIER TOULOUSE

DIRECTION DE L'UNIVERSITE

M. LARENG.	Président
M. MATHIS	1er Vice-Président
M. MARTIN	2ème Vice-Président

SCIENCES

HONORARIAT

M. BLAIZOT	Doyen honoraire
M. CAPDECOMME	Doyen honoraire, Recteur honoraire
M. DELTHEIL	Doyen honoraire, Recteur honoraire, Professeur honoraire
M. DUPOUY	Membre de l'Institut, Doyen honoraire, Directeur Général honoraire du C. N. R. S. , Professeur honoraire
M. DURAND	Doyen honoraire
M. FOUCHE	Professeur honoraire
M. GAUSSEN	Professeur honoraire, Correspondant de l'Institut
M. JACQUES	Professeur honoraire
M. MATHIS	Doyen honoraire
M. MIGNONAC	Professeur honoraire
M. MORQUER	Professeur honoraire, Correspondant de l'Institut
M. SECONDAT	Professeur honoraire
M. TEISSIE-SOLIER	Professeur honoraire
M. VANDEL	Professeur honoraire, Membre de l'Institut

PROFESSEURS

M. ESCANDE	Mécanique des Fluides, Membre de l'Institut
M. CASTERAS	Géologie, Correspondant de l'Institut
M. BEDOS	Chimie
M. CAPDECOMME	Minéralogie
M. NOUVEL	Biologie Générale
M. GALLAIS	Chimie, Correspondant de l'Institut
M. DURAND	Physique
M. FERT	Physique
M. DIEHL	Agronomie
M. LESBRE	Chimie Organique
M. BRUNEL	Physiologie Végétale
M. HURON	Mathématiques Appliquées
M. MARGULIS	Pédologie

M. TRICHE	Chimie Analytique et Spectrographie
M. LEDOUX	Zoologie Appliquée
M. RIVALS	Agriculture
M. PERRIER	Physique
M. MATHIS	Chimie
M. ORLIAC	Géochimie et Minéralogie
M. LAFOURCADE	Physique
M. ANGELIER	Zoologie
Melle de FERRE	Botanique
M. FARRAN	Minéralogie et Géotechnique
M. LAUDET	Physique Théorique et Calcul Numérique
M. SERFATY	Biologie animale
M. NOUGARO	Hydraulique Générale et Appliquée
M. LAGASSE	Electrotechnique
M. GARDY	Génie Chimique
M. BLANC	Physique Nucléaire
M. BIREBENT	Electronique Appliquée
M. VOIGT	Chimie Minérale
M. LEREDDE	Botanique
M. LELUBRE	Géologie
M. TROCHAIN	Botanique
M. LALAGUE	Mathématiques Générales
M. BOUIGUE	Astronomie
M. COUCHET	Physique
M. LEJEUNE	Chimie
M. LASCOMBES	Botanique
M. ASSELINEAU	Chimie Biologique
M. MAURET	Chimie Systématique
M. AGID	Physiologie
M. MONTANT	Cryptogamie
M. GAUTIER	Physique
M. CRUMEYROLLE	Mathématiques
M. GOURINARD	Géologie
M. PULOU	Minéralogie
M. CAMBOU	Physique Spatiale
M. MIQUEL	Chimie
M. LACOSTE	Electrotechnique
M. THIBAULT	Mécanique Rationnelle et Appliquée
M. MONTEL	Chimie Physique et Electrochimie
M. MASCART	Mathématiques
M. MEDIONI	Psychophysiologie
M. THIRRIOT	Hydrodynamique E.N.S.E.I.H.T.
M. BRENON	Géologie
M. LAFON	Calcul Différentiel et Intégral
M. RAYNAUD P.	Physiologie Animale
M. ZALTA	Chimie Biologique
M. SEVELY	Electrotechnique
M. POMMIEZ	Mathématiques
M. REY	Biologie Végétale
M. COULOMB	Physique

M. TRINQUIER	Physique
M. MARONI	Chimie
M. BEETSCHEN	Biologie Générale
M. GRUAT	Mécanique des Fluides (Hydraulique)
M. DERACHE	Physiologie Animale

PROFESSEUR ASSOCIE

M. SAKUMA

PROFESSEUR SANS CHAIRE

M. HAMANT	Botanique
Mme BRUNEL	Physiologie Végétale
Mlle BERDUCOU	Physiologie Végétale
M. PICCA	Physique
M. MERIC	Mathématiques Appliquées
Mme LECAL	Zoologie
M. BUGAREL	Génie Chimique
M. MIROUSE	Géologie
M. PILOD	Physique
M. LARROQUE	Physique
M. BARRANS	Chimie
Mme LAFON	Mathématiques
Mlle LAPEYRE	Mathématiques - Informatique
M. DAT	Mécanique des Fluides
M. JEANNIN	Chimie Minérale
M. BERTRAND	Chimie
M. SATGE	Chimie Organique
M. DESQ	Mathématiques
M. MAHENC	Chimie
M. LATTES	Chimie
M. ROCARD	Electronique
M. DEGEILH	Physique
M. FALLOT	Biologie Végétale (Viticulture et Arboriculture)
M. POILBLANC	Chimie
M. TOUZE	Physiologie Végétale
M. FRASNAY	Mathématiques
M. CASSAGNAU	Zoologie
M. GUERIN	Mathématiques
M. TRUCHASSON	Mécanique
M. PESCIA	Physique
M. HOFFMANN	Electronique
M. SOUQUET	Géologie
M. SCHNEIDER	Biologie cellulaire
M. SICARD	Biologie génétique
M. DE LOTH	Chimie-Physique
M. ENJALBERT	Chimie
M. MARTIN J.C.	Génie Electrique

MAITRES DE CONFERENCES

M. BLANCHARD	Mathématiques
M. BITSCH	Zoologie
M. FONTAN	Physiologie Nucléaire
M. CAUSSINUS	Mathématiques Appliquées
M. MARTY	Physique Industrielle
M. BAUDRAS	Chimie Biologique
M. DURAND Ph.	Physique
M. LEFEUVRE	Electronique E.N.S.E.I.H.T.
M. SAPORTE	Physique
M. ATTEIA	Mathématiques
M. IMBERT	Psychophysiologie
M. ANGELINO	Génie Chimique
M. THENOZ	Génie Civil
M. DABOSI	E.N.S.C.
M. CALVET	Mécanique des Fluides (E.N.S.E.I.H.T.)
M. PAGANI	Physique
M. CUPPENS	Mathématiques
M. PERENNOU	Informatique
M. REY	Génie Electrique
M. MATHIEU	Chimie Analytique
M. TRANNOY	Physique (E.N.S.E.I.H.T.)
M. CASTAGNETTO	Mathématiques (E.N.S.E.I.H.T.)
M. CASTAN	Informatique
M. GILOT	Génie Chimique (Institut du Génie Chimique)
M. LANTELME	Chimie Industrielle
M. CALMON	Techniques Agricoles
Mlle BARBANCE	Mathématiques
M. VERDIER	Physique
M. GILLY	Génie Mécanique
M. VEDRENNE	Mesures Physiques
M. BEAUFILS	Informatique
M. MATHEAU	Electronique (E.N.S.E.I.H.T.)
M. COTTU	Génie Mécanique
M. ROUTIE	Génie Chimique
M. REME	Mesures Physiques
M. MARAL	Physique

MAITRES DE CONFERENCES ASSOCIES

M. PUN

CHARGES D'ENSEIGNEMENT

M. MAUSS	Mécanique
M. PRADINES	Mathématiques
M. BARTHET	Mathématiques

PERSONNEL DE L'UNIVERSITE AFFECTE A L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES DE TOULOUSE.

PROFESSEURS

M. ROQUES	Chimie Industrielle
M. GRATELOUP	Génie Electrique
M. MASO	Génie Civil
M. MIRA	Génie Electrique

PROFESSEURS SANS CHAIRE

M. CASSIGNOL	Génie Electrique
M. COLLETTE	Physique
M. CARRARA	Physique

MAITRES DE CONFERENCES

M. FAGET	Physique
M. SCHUTTLER	Physique
M. LETURCQ	Génie Electrique
M. SIRIEYS	Génie Civil
M. DURAND G.	Chimie

CHARGES D'ENSEIGNEMENT

M. BROUSSEAU	Physique
M. LACAZE	Mathématiques

ADMINISTRATION

M. PRINEAU	Secrétaire Général de l'Université Paul Sabatier

A la mémoire de mon père.

A ma femme.
A notre fille.
A tous les miens.

Avec ma profonde affection.

AVANT - PROPOS

Nous présentons, dans ce mémoire, une synthèse des études que nous avons développées dans le cadre du Laboratoire d'Automatique et de ses Applications Spatiales dirigé par Monsieur le Professeur LAGASSE.

A cette occasion, nous tenons à exprimer notre très vive gratitude à Monsieur le Professeur LAGASSE pour nous avoir accepté dans son Laboratoire et nous avoir fourni l'environnement et les moyens propices à la réalisation de notre travail. Son attitude est et a toujours été un exemple précieux dans la voie que nous avons choisie.

Nos remerciements vont également à Monsieur le Professeur GRATELOUP, Directeur du Département de Génie Electrique de l'Institut National des Sciences Appliquées dont les remarques nous ont permis de mieux parachever ce mémoire.

Nous sommes particulièrement sensibles à l'honneur que nous fait le Professeur FLORINE en acceptant de siéger à notre Jury. L'étendue de ses connaissances dans le domaine de la logique le désignait tout particulièrement pour cette tâche. Qu'il trouve ici l'expression de notre gratitude.

Monsieur Pierre AZEMA, Chargé de Recherche au C.N.R.S. a suivi ce travail depuis ses débuts. Il a partagé nos difficultés quotidiennes en conjuguant l'autorité d'un maître et la sollicitude d'un ami ; puissent, ces quelques mots, lui exprimer notre reconnaissance.

Cette étude n'aurait pu se développer sans le support d'un Laboratoire. Nous sommes heureux en cette circonstance, de rendre hommage à tout le personnel qui, à titre divers, assure la vie du Laboratoire d'Automatique et de ses Applications Spatiales.

Nous mentionnerons de façon plus particulière : les chercheurs de l'équipe "Structures Logiques" pour l'aide et les encouragements qu'ils nous ont apportés. Monsieur J.E. DOUCET, Ingénieur au Service "Informatique et Simulation" pour son ingéniosité dans le développement des programmes.

Le Service de Documentation, dirigé par Madame IPPOLITO, qui a mis en forme le mémoire et où Monsieur LAPEYRE-MESTRE, Mademoiselle FIERT, Monsieur FRESPUECH n'on pas ménagé leur peine.

INTRODUCTION

La logique cellulaire est une branche de la théorie de la commutation qui traite des techniques d'analyse et de synthèse des réseaux logiques ayant la forme d'arrangements cellulaires.

Un arrangement cellulaire est un arrangement itératif à une, deux ou trois dimensions de cellules similaires ou identiques possédant une structure d'interconnexion entre cellules uniforme. Une cellule est une combinaison d'éléments connectables de façon à réaliser une ou plusieurs fonctions logiques. Les éléments sont des ensembles de composants physiques élémentaires actifs ou passifs tels que transistors, diodes, résistances, etc.

Des circuits logiques ont déjà été réalisés dans le passé qui exhibaient cette propriété de "cellularité" tels les réseaux décodeurs à diodes ou les matrices de fonctions dans les premiers calculateurs ; mais la cellularité de ces circuits était plus un moyen commode de représentation spatiale que la réflexion d'une tendance dans la technologie.

Les études sur les réseaux logiques cellulaires ont pris une importance nouvelle au début des années 1960 [1] avec l'avènement de la technologie des circuits intégrés. Dans cette technologie un élément logique individuel peut contenir un grand nombre de composants actifs ou passifs interconnectés sur un substrat monolithique appelé "puce". De plus la technologie de l'intégration à large échelle permet de produire et d'interconnecter un nombre important de puces sur une plaquette. Nous pensons qu'il est économiquement intéressant pour un fabricant de produire une plaquette sous la forme d'un arrangement cellulaire de puces. En effet, outre son prix de revient, une fabrication de ce type présente plusieurs avantages. D'abord la densité d'assemblage est plus élevée non seulement par la réduction de taille des composants individuels mais aussi par l'élimination de beaucoup de fils d'interconnexion entre puces dont l'emploi est courant dans la production classique de modules fonctionnels composés de boîtiers de circuits intégrés. Les pannes dues à des défauts sur les fils d'interconnexion, qui sont assez fréquentes

dans les circuits logiques classiques, sont ainsi réduites dans de grandes proportions.

Une des caractéristiques essentielles des circuits cellulaires réside, à notre avis, dans leur souplesse au niveau des performances. Cette souplesse résulte d'une possibilité de modification soit dans la fonction de chaque cellule de l'arrangement, soit dans la structure d'interconnexion. La spécialisation des réseaux s'obtient de plusieurs manières : par exemple par la mise en place de paramètres d'interconnexion qui ne sont assignés qu'aux étapes finales de la production (circuits "à la demande du client") ou par l'association à chaque cellule d'un ensemble de paramètres qui pourront être programmés électroniquement par des signaux de commande appliqués aux extrémités du réseau. Nous qualifierons de programmables ces réseaux [2].

Depuis 1960 de très nombreuses études [3, 4, 5, 6] ont porté sur les réseaux cellulaires et ont permis de dégager les principaux problèmes posés par l'analyse et la synthèse de ces réseaux. Parmi ceux-ci citons :

1 - Choix de la structure d'interconnexion : nous nous limiterons dans ce mémoire à l'étude de réseaux possédant une structure d'interconnexion très simple mais se révélant être potentiellement très utile. Ce type de structure est représenté sur la figure I.

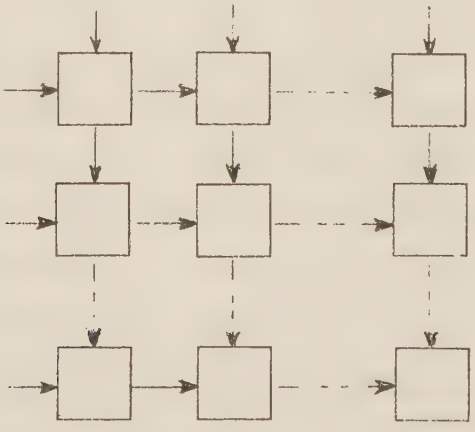


Figure I - Structure d'interconnexion d'un réseau cellulaire

2 - Choix de la cellule de base : en fait il n'existe pas de critère permettant de juger à priori de l'efficacité logique de tel ou tel opérateur dans un réseau cellulaire. Nous présenterons ici les résultats obtenus avec plusieurs types d'opérateurs.

3 - Algorithmes de synthèse pratiques et utilisables : c'est dans ce domaine que nos efforts se sont surtout portés. Nous pensons y avoir obtenu une réussite au moins partielle dans le mesure où nous présentons plusieurs méthodes de synthèse systématique de fonctions logiques arbitraires à l'aide de réseaux cellulaires.

Malgré leurs avantages les réseaux cellulaires n'ont pas encore reçu l'acceptation universelle des constructeurs de circuits logiques. Les problèmes dont nous venons de parler ainsi que certaines difficultés au niveau du test et de la maintenance expliquent cette réticence. Les fabricants certes expérimentent sur la logique cellulaire : c'est ainsi que nous voyons apparaître sur le marché des réseaux cellulaires simples réalisant la fonction multiplication ou la fonction multiplexage. La tendance la plus générale reste cependant encore au niveau de l'assemblage en circuits intégrés de types de cellules standard avec des interconnexions métallisées à la demande. Cette attitude évoluera certainement dans la mesure où il est probable que quelques uns des problèmes techniques majeurs de l'intégration à large échelle seront résolus dans les prochaines années. De plus, la poursuite des études théoriques sur les réseaux cellulaires permet une meilleure compréhension de leur comportement qui doit rejaillir au niveau de la fabrication : c'est dans ce cadre que se situe notre travail.

Dans un premier chapitre nous donnons un bref aperçu des principales recherches sur les réseaux cellulaires et des problèmes qu'elles soulèvent.

Dans le deuxième chapitre nous présentons deux méthodes de synthèse de réseaux basées sur l'adaptation de propriétés booléennes classiques l'une s'appliquant aux réseaux en cascade l'autre aux réseaux rectangulaires bidimensionnels.

Enfin dans le troisième chapitre nous développons une méthode spécifique d'étude des réseaux cellulaires. Cette méthode fait appel aux propriétés des éléments à seuil qui permettent de traduire simplement le fonctionnement d'un réseau à partir d'un système d'inégalités linéaires. La résolution de ce système se ramène à la résolution d'un problème de programmation linéaire en variables bivalentes pour laquelle nous proposons deux algorithmes.

CHAPITRE I

DESCRIPTION DE QUELQUES METHODES ET TECHNIQUES DE LA LOGIQUE CELLULAIRE

I-1 - LOGIQUE CELLULAIRE ET INTEGRATION A LARGE ECHELLE

Après avoir donné un bref aperçu des différentes approches technologiques à l'intégration à large échelle nous essaierons dans ce paragraphe de les relier à la logique cellulaire et de montrer l'intérêt que présente cette dernière dans le développement de cette technique particulière.

I-1-1 - Evolution des techniques de fabrication de circuits intégrés: intégration à large échelle

Un des évènements les plus importants de la dernière décennie dans le domaine des circuits logiques concerne l'avènement sur le marché de composants intégrés pratiques, fiables et économiques. Alors que dans un passé très récent les réalisateurs de systèmes logiques manipulaient des éléments individuels tels que transistors, résistances ou capacités, la technologie de fabrication s'est développée à un point tel qu'ils disposent maintenant d'un grand nombre de composants interconnectés sur un substrat monolithique. Les courbes prospectives laissent à penser que cette tendance ne peut que s'affirmer et il est raisonnable de prévoir un nombre de 10^5 éléments par puce dans un avenir très proche. Cette intégration à large échelle pose le problème de savoir comment connecter ces éléments pour réaliser des sous-systèmes ou des systèmes logiques complets. A l'heure actuelle il existe essentiellement deux approches pour résoudre ce problème :

a - Intégration ayant pour base le composant physique : des équipements intégrés complets sont fabriqués en incorporant un nombre important d'éléments sur une surface donnée de silicium. Le réalisateur se trouve dans ce cas confronté avec le problème de connecter de façon optimale un ensemble de composants physiques intégrés pour faire une fonction ou un système logique.

S'il veut par exemple, réaliser un registre à décalage à MOST, il connectera des MOST et des résistances de façon optimale sans essayer de partitionner le registre en plusieurs blocks de base. Cette approche permet d'obtenir une très grande densité d'éléments par unité de surface ou ce qui revient au même de réaliser un équipement intégré complexe sur une très petite surface. De plus le temps de propagation des signaux est faible. Par contre la grande densité d'éléments limite le nombre de points de tests disponibles ; la durée de réalisation est importante, chaque système intégré nécessite un ensemble complet de masques y compris ceux de diffusion et de métallisation. Ce procédé n'est acceptable que pour des produits standard ou pour des réalisations à la demande d'un volume important.

b - Intégration ayant pour base la cellule : des équipements intégrés complets sont élaborés en partant de cellules qui réalisent des fonctions logiques élémentaires. Le client fournit des équations booléennes et les spécifications de l'équipement qu'il désire. Le constructeur adopte alors l'une ou l'autre des procédures suivantes :

- soit il dispose de pastilles de silicium traitées sur lesquelles sont intégrées par exemple un nombre important de portes ON. Ces pastilles sont munies d'une première couche de métallisation et il suffit alors d'élaborer les masques qui permettront d'interconnecter les portes d'une certaine manière.

- soit il dispose de pastilles sur lesquelles est intégré un ensemble de cellules. Chaque cellule de base a des éléments suffisamment nombreux pour que plusieurs circuits logiques différents puissent être obtenus par variation de la première couche de métallisation.

Outre la souplesse dans la technique de fabrication cette approche présente l'avantage d'aboutir à une structure d'interconnexion régulière. De plus, la conception du système intégré est facilitée par le fait que les équations logiques booléennes s'expriment aisément en termes de ON, NI, etc. L'inconvénient principal réside dans l'utilisation d'une surface de silicium importante.

I-1-2 - Intérêt de la logique cellulaire dans l'intégration à grande échelle

Les méthodes et techniques de la logique cellulaire sont tout à fait adaptées à la deuxième approche de l'intégration à large échelle que nous avons décrite ci-dessus. La logique cellulaire constitue en quelque sorte une tentative de reformulation des procédures booléennes pour les rendre compatibles avec les nouvelles techniques de fabrication. Un réseau cellulaire est formé d'un ensemble de cellules ; par sa conception même chaque cellule dispose ou non de moyens qui permettent de la modifier afin de pouvoir réaliser une fonction logique parmi un catalogue donné de fonctions. Ces moyens sont les paramètres de la cellule. Ainsi pour plusieurs choix appropriés mais fixes des connexions entre cellules il est possible de spécifier les paramètres de telle sorte que n'importe quelle propriété fonctionnelle désirée puisse être obtenue à partir d'un réseau cellulaire à condition de choisir convenablement ses dimensions.

Les procédures de réalisation d'une fonction logique donnée peuvent être classées en deux catégories :

- les paramètres de la cellule sont utilisés uniquement pour modifier la structure d'interconnexion mais non la fonction de la cellule

- les paramètres de la cellule sont utilisés à la fois pour modifier la structure d'interconnexion et la fonction de la cellule

Dans la réalisation pratique des cellules les paramètres seront soit des métallisations, soit des bascules commandées par des signaux logiques identiques aux variables d'entrée.

I-2 - METHODES ET TECHNIQUES DE LA LOGIQUE CELLULAIRE

Nous distinguerons deux grandes classes de réseaux cellulaires :

- les réseaux non spécialisés ou systématiques qui ont un comportement logique arbitraire

- les réseaux spécialisés ou "à la demande" qui réalisent des fonctions logiques particulières

I-2-1 - Réseaux cellulaires non spécialisés

I-2-1-1 - Cascade de Maitra [7]

Ce sont les plus simples des structures cellulaires. La cellule de base C est un élément logique avec deux entrées x et y et une sortie $z = f(x, y)$ où f est une des seize fonctions booléennes de deux variables. La figure I-1 représente une telle cellule.

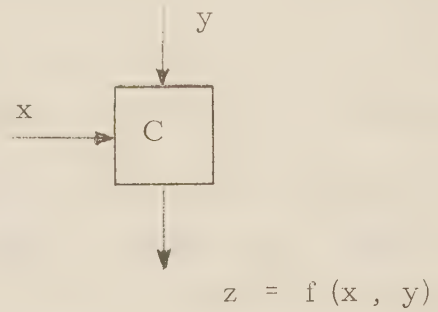


Figure I-1 - Cellule de Maitra

Une cascade cellulaire est une structure composée de m cellules C_i, $1 \leqslant i \leqslant m$, telle que la sortie de la $i^{ième}$ cellule z_i soit connectée à l'entrée de la cellule $i + 1 : x_{i+1}$. La figure I-2 représente une telle cascade.

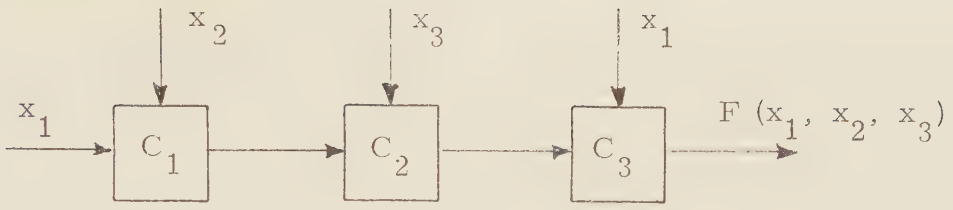


Figure I-2 - Cascade de Maitra redondante

Les connexions entre cellules sont appelées connexions artérielles. Pour une cascade de m cellules, il y a m + 1 entrées externes (en comptant les m entrées verticales et l'entrée horizontale unique). Si ces entrées sont connectées à m + 1 variables booléennes distinctes la cascade est dite irrédondante, tandis que si une ou plusieurs entrées sont attaquées par la même variable booléenne la cascade est dite redondante (figure 3). La caractéristique principale de ces réseaux est qu'ils sont fonctionnellement incomplets, c'est-à-dire qu'il existe des fonctions booléennes qui ne peuvent être réalisées par de tels réseaux. On montre que la fraction de fonctions de n variables réalisables par ces cascades devient très faible lorsque n croît. Un problème intéressant dans l'étude de ces structures est celui de la mise au point d'algorithmes permettant de tester la réalisabilité d'une fonction. Nous reviendrons plus en détails sur ce problème dans la première partie du chapitre II. Nous donnons simplement ici un tableau permettant d'évaluer les performances de ce type de réseau.

Nombre de variables n	2	3	4
Nombre de classes réalisables m	m =	m =	m =
a. Cascade irrédondante	5	14	35
b. Cascade redondante	5	19	135
Nombre total de classes M	5	22	402
Nombre de fonctions réalisables p	p =	p =	p =
a. Cascade irrédondante	16	152	2368
b. Cascade redondante	16	242	21216
Nombre total de fonctions N	16	256	65536

Tableau I - Performances des cascades de Maitra

I-2-1-2 - Cascades de Short. Cascade double rail [8]

Il y a plusieurs moyens d'accroître les possibilités logiques d'une cascade :

1) augmenter le nombre d'entrées externes de chaque cellule

2) augmenter le nombre de connexions entre cellules

3) combiner les méthodes 1) et 2).

Toutes ces approches entraînent une complexité supplémentaire de la fonction de la cellule et nous pouvons donc penser intuitivement qu'elles augmentent le nombre de fonctions réalisables.

Lendaris et Stanley ont étudié la première de ces approches. Ils supposent que chaque cellule possède m connexions externes et une seule connexion artérielle et qu'elle peut réaliser sur sa sortie toutes les fonctions de m + 1 variables. Bien que le nombre de fonctions réalisables par une telle cascade soit important (en particulier toutes les fonctions de m + 1 variables) elle ne les réalise cependant pas toutes. Bien plus ce défaut empire lorsque le nombre de variables croît.

La seconde approche basée sur une augmentation du nombre d'interconnexions entre cellules constitue une généralisation des cascades de Maitra décrites dans la section précédente. Nous donnons sur la figure I-3 un exemple d'une telle cascade pour des fonctions de quatre variables. Nous supposerons que la cellule est assez complexe pour produire toute fonction de ses trois entrées. Cette cascade est logiquement complète. Sa souplesse plus grande est illustrée par la structure de la figure I-4 qui réalise toute fonction de quatre variables.

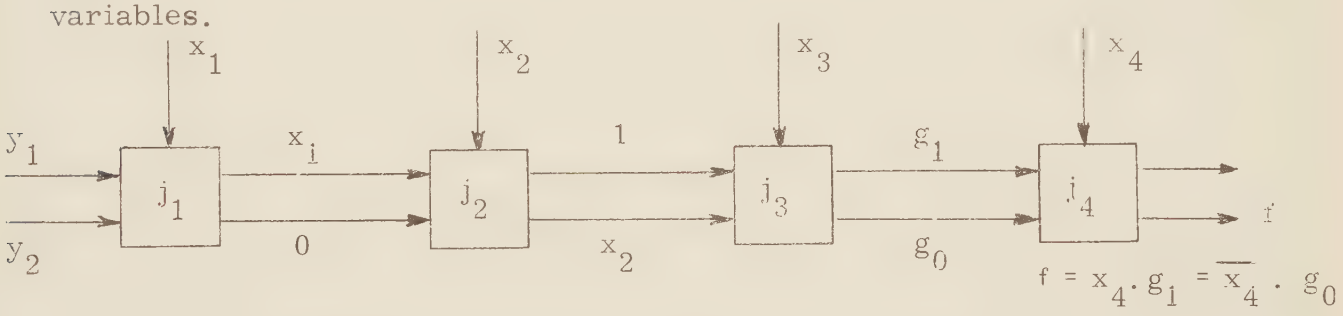


Figure I-3 - Cascade double rail (fonction de 4 variables)

La spécialisation de la cellule j sera notée :

$$j : f_1 (x , y_1, y_2) , f_2 (x , y_1, y_2)$$

où f_1 et f_2 sont les sorties, y_1 et y_2 les entrées artérielles et x l'entrée primaire. Si nous prenons :

$$j_1 : x_1, 0$$

$$j_2 : 1, x_2$$

$$j_3 : g_1 (x, y_1, y_2) , g_o (x, y_1, y_2)$$

$$j_4 : \text{arbitraire, } x y_1 + \overline{x} y_2$$

où g_o et g_1 sont des fonctions arbitraires de trois variables, alors la cascade de la figure I-3 réalisera toute fonction de quatre variables qui peut être mise sous la forme :

$$f (x_1, x_2, x_3, x_4) = x_4 g_1 + \overline{x}_4 g_o$$

sur la sortie du bas de la dernière cellule.

Le fait que ce type de cascade soit logiquement complet peut être prouvé de la façon suivante : supposons que la fonction à réaliser soit spécifiée sous forme de somme de produits ; les deux sorties de la cellule sont données par :

$$f_1 = (x \ y_1, \overline{x} y_1, 1)$$

$$f_2 = (x y_1 + y_2, \overline{x} y_1 + y_2, y_2)$$

Nous pouvons alors former un terme produit arbitraire sur f_1 et l'ajouter à la somme accumulée sur f_2. Au même point nous pouvons commencer un nouveau terme produit sur f_1. Evidemment la cascade résultante sera redondante. La figure I-4 illustre une telle cascade

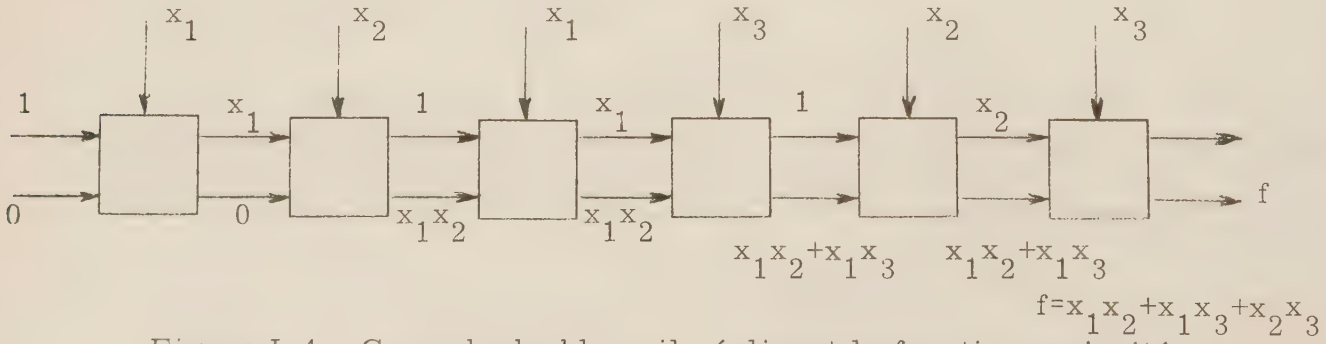


Figure I-4 - Cascade double rail réalisant la fonction majorité de 3 variables

Trois problèmes essentiels se posent lors de la réalisation d'une fonction par une cascade double rail. Le premier est la recherche d'une limite supérieure sur le nombre de cellules d'une cascade double rail universelle, c'est-à-dire une cascade réalisant n'importe quelle fonction arbitraire $f(x_1, \ldots, x_n)$. Le second concerne le développement d'algorithmes pour la synthèse d'une fonction donnée sur une cascade de longueur minimale. Ce problème n'est pas résolu. Enfin, il serait intéressant de trouver un ensemble de fonctions réduit sinon minimum que la cellule de base d'une cascade universelle doit produire.

I-2-1-3 - Réseaux bidimensionnels

Ce paragraphe décrit quelques uns des arrangements cellulaires connus utilisés pour la réalisation de fonctions logiques combinatoires arbitraires. Une classe intéressante de réseaux est obtenue à partir d'une structure d'interconnexion fixe et est illustrée sur la figure I-5.

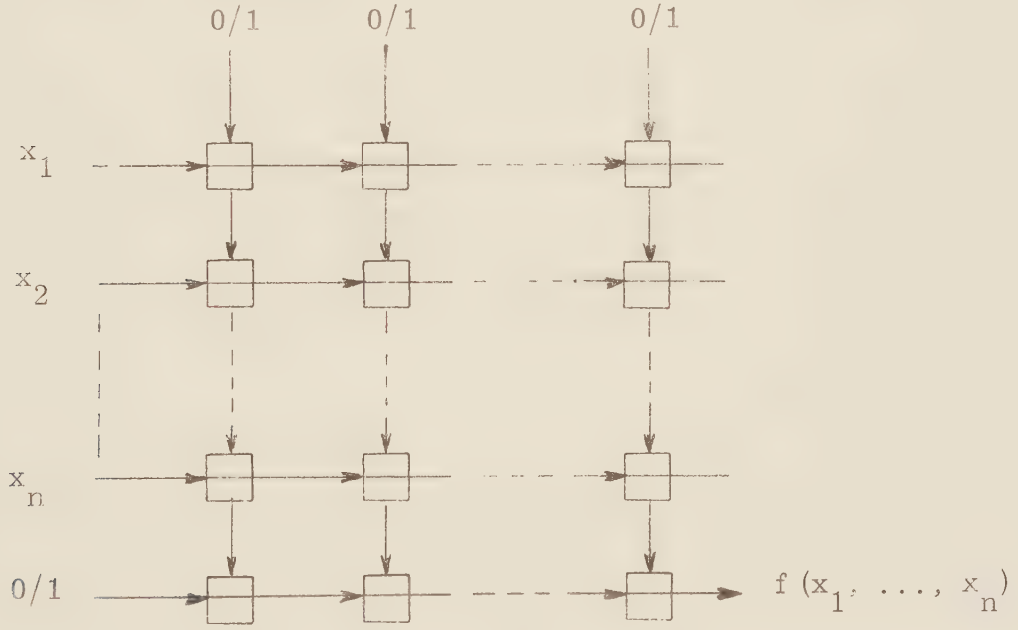


Figure I-5 - Réseau cellulaire à structure d'interconnexion particulière

Chaque cellule du réseau principal réalise une fonction à deux entrées et une sortie. L'entrée la plus haute de chaque colonne de cellules est une constante 0 ou 1. La ligne de cellules du bas est une ligne collectrice qui est aussi une cascade de cellules à deux entrées et une sortie, l'entrée la plus à gauche étant une constante 0 ou 1. La sortie de la ligne collectrice est la fonction désirée $f(x_1, \ldots, x_n)$ des variables d'entrées $x_1, \ldots x_n$. Afin d'exploiter quelques uns des développements canoniques d'une fonction logique pour les algorithmes de synthèse, nous supposerons que les cellules de la ligne collectrice réalisent la même fonction OU, ou ET, ou OU EXCLUSIF, ou EQUIVALENCE.

I-2-1-3-1 - Réseaux à points de coupure [9]

Ces réseaux sont construits à partir de cascades de Maitra. Chaque colonne de la figure I-5 peut produire toutes les fonctions réalisables par une cascade de Maitra. Nous supposerons que les cellules de la ligne collectrice réalisent des fonctions OU ou ET. Toute fonction exprimée comme une somme de produits (produit de sommes) peut être réalisée en obtenant les produits (sommes) par les colonnes et leur somme (produit) par la ligne collectrice de cellules OU (ET). Donc pour toute fonction arbitraire de n variables, $n + 1$ lignes et un maximum de 2^{n-1} colonnes donc un maximum de $(n + 1)\, 2^{n-1}$ cellules à 2 entrées et une sortie sont nécessaires. Le nombre de cellules représentera le taux de croissance du réseau.

Bien que chaque colonne puisse produire toutes les fonctions réalisables par une cascade de Maitra, il n'est pas nécessaire que les cellules individuelles produisent les seize fonctions de leurs deux entrées. Cette propriété résulte du théorème suivant que nous énoncerons sans démonstration :

Théorème 1 : Si f (x_1, ..., x_n) peut être réalisée par une cascade de Maitra de n cellules dans laquelle l'entrée de la cellule i est x_i ($1 \leq i \leq n$) alors f peut aussi être réalisée par une cascade dont l'entrée de la cellule est x_i et qui produit simplement les six fonctions suivantes :

1 - $f_1 = \overline{x_i} \cdot \overline{y_i}$ (NI) ou $f_2 = \overline{x_i} \cdot y_i$ (Non implication)

2 - $f_4 = x_i \cdot \overline{y_i}$ (Non implication) ou $f_8 = x_i \cdot y_i$ (ET)

3 - $f_5 = \overline{y_i}$ (PAS) ou $f_{10} = y$

4 - $f_6 = x_i \oplus y_i$ (OU EXCLUSIF) ou

$f_9 = x_i \oplus \overline{y_i}$ (EQUIVALENCE)

5 - $f_7 = \overline{x_i} + \overline{y_i}$ (ON) ou $f_{11} = \overline{x_i} + y_i$ (IMPLICATION)

6 - $f_{13} = x_i + \overline{y_i}$ (IMPLICATION) ou $f_{14} = x_i + y_i$ (OU)

où y_i est l'entrée artérielle de la cellule et f_j est la $j^{ème}$ fonction de deux variables dont l'image caractéristique a un équivalent décimal de valeur j.

Ainsi la cellule de Maitra réduite doit être capable de produire un ensemble de six fonctions et nécessite donc au moins trois bits de spécification ou "points de coupure". Dans la réalisation physique de la cellule ces points de coupure prennent la forme de cellules photoélectriques ou de commutateurs ou simplement de points auxquels la connexion électrique peut être interrompue à l'étape finale spécialisant aussi le réseau pour une fonction donnée. Dans la cellule initiale étudiée par Minnick un quatrième bit de spécification était inclus pour permettre la conversion de la cellule en un élément mémoire. Minnick a décrit un ensemble d'éléments de base de calculateurs

tels que décodeurs, complémenteurs, translateurs de code, registres utilisant ces cellules. Cet auteur a aussi proposé des méthodes de réalisation logique pour l'emploi de ces réseaux mais excepté quand le réalisateur applique ses propres ingénuité et savoir faire, ces méthodes s'avèrent inefficaces à cause en particulier du nombre élevé de cellules qui réalisent peu de fonctions logiques utiles.

I-2-1-3-2 - Réseaux additionneurs [10]

Un autre réseau intéressant du type décrit sur la figure I-5 est construit à partir d'une cellule qui peut produire soit le ET soit le OU de ses entrées suivant la valeur d'un seul bit de spécification b. La sortie z_i de la cellule i dont l'entrée est x_i peut s'écrire :

$$z_i = x_i \cdot y_i + b_i \cdot y_i + b_i \cdot x_i$$

où y_i est l'entrée artérielle et b_i vaut 0 ou 1. Chaque cellule de la cascade collectrice réalise la fonction OU EXCLUSIF. Nous voyons que la fonction z_i représente aussi la sortie de report d'un additionneur complet à un seul bit dont b_i serait la troisième entrée. La fonction produite par une telle cascade est appelée fonction q et est notée q_i si $(b_n \; b_{n-1} \ldots b_2 \; b_1)$ est l'équivalent binaire de l'entier positif i $(0 \leqslant i \leqslant 2^n - 1)$. Les fonctions q pour n = 2 sont représentées sur la table II.

Chaque fonction q est une fonction monotone positive et possède un seul groupe de 1 dans son image caractéristique et ceci est vrai pour tout n.

b_2 b_1	Vecteur caractéristique c_3 c_2 c_1 c_0				Fonctions q
0 0	0	0	0	0	$q_0 = 0$
0 1	1	0	0	0	$q_1 = x_1 x_2$
1 0	1	1	0	0	$q_2 = x_2$
1 1	1	1	1	0	$q_3 = x_1 + x_2$

Table II - Fonctions q pour n = 2

Tout produit fondamental

$$p_i = \dot{x}_n \, \mathring{x}_{n-1} \cdots \dot{x}_2 \, \mathring{x}_1 \quad , \quad i = \sum_{j=1}^{n} f_j \; 2^{j-1}$$

où

$$f_j = \begin{cases} 1 \; , & \text{si} \quad \dot{x}_j = x_j \\ 0 \; , & \text{si} \quad \dot{x}_j = \overline{x_j} \end{cases}$$

peut se mettre sous la forme :

$$p_i = q_{2^n - i} \oplus q_{2^n - i - 1}$$

où q_{2^n} est définie comme toujours vraie. Toute fonction arbitraire peut être spécifiée comme : $(0 \leqslant i \, , \, j, \ldots, \, k \leqslant 2^{n-1})$

$$f = p_i \oplus p_j \oplus \ldots \oplus p_k$$

$$= q_{2^n-i} \oplus q_{2^n-i-1} \oplus q_{2^n-j} \oplus q_{2^n-j-1} \oplus \ldots$$

$$\ldots \oplus q_{2^n-k} \oplus q_{2^n-k-1}$$

Ainsi, nous pouvons faire la synthèse d'une fonction arbitraire en employant un ensemble avec une ligne collectrice qui forme le OU EXCLUSIF des sorties de cascades. De plus ,

$$p_i + p_{i+1} = q_{2^n-i} \oplus q_{2^n-i-1} \oplus q_{2^n-i-1} \oplus q_{2^n-i-2}$$

$$= q_{2^n-i} \oplus q_{2^n-i-2}$$

et en général

$$p_i + p_{i+1} + \ldots + p_{i+k} = q_{2^n-i} \oplus q_{2^n-(i+k)-1}$$

Ainsi nous comptons les groupes de bits semblables dans l'image caractéristique et si n est le nombre de ces groupes, alors la réalisation d'une fonction par un réseau additionneur nécessite n - 1 colonnes.

Exemple : $f = x_3 \overline{x}_2 \overline{x}_1 + \overline{x}_3 (x_1 + x_2) + x_1 x_2$

L'image caractéristique de f est $(c_7\ c_6\ c_5\ c_4\ c_3\ c_2\ c_1\ c_0) = (1\ 0\ 0\ 1\ 1\ 1\ 1\ 0)$. Il y a quatre groupes de bits semblables (1, 0 0, 1 1 11, 0). Nous avons $p_1 + p_2 + p_3 + p_4 = q_7 \oplus q_3$ et $p_7 = q_1$. De plus $f = q_7 \oplus q_3 \oplus q_1$.

Les bits de spécifications pour q_7, q_3 et q_1 sont respectivement ($b_3 = 1$, $b_2 = 1$, $b_1 = 1$), ($b_3 = 0$, $b_2 = 1$, $b_1 = 1$) et ($b_3 = 0$, $b_2 = 0$, $b_1 = 1$).

Le taux de croissance d'un réseau additionneur peut être estimé à partir des travaux d'Elspas et Short [11] sur la mesure du nombre de groupes de bits semblables pour les fonctions logiques. Ces auteurs ont montré que le nombre maximum de groupes de bits r (f) pour une fonction arbitraire de n variables est $(2^{n+1}+1)/3$ quand n est pair et $(2^{n+1} + 2)/3$ quand n est impair. Ceci signifie que le taux de croissance d'un réseau additionneur est (n + 1) r (f) quand des permutations et des complémentations arbitraires des variables d'entrée sont permises.

Les réseaux additionneurs peuvent être simplifiés en combinant certaines fonctions q de la même façon que des produits fondamentaux ou mintermes peuvent être combinés pour donner des implicants premiers.

I-2-1-3-3 - Réseaux à fonctions symétriques

Un réseau qui réalise les fonctions symétriques élémentaires S_0, S_1, ..., S_n des variables x_1, x_2, ..., x_n est montré sur la figure I-6.

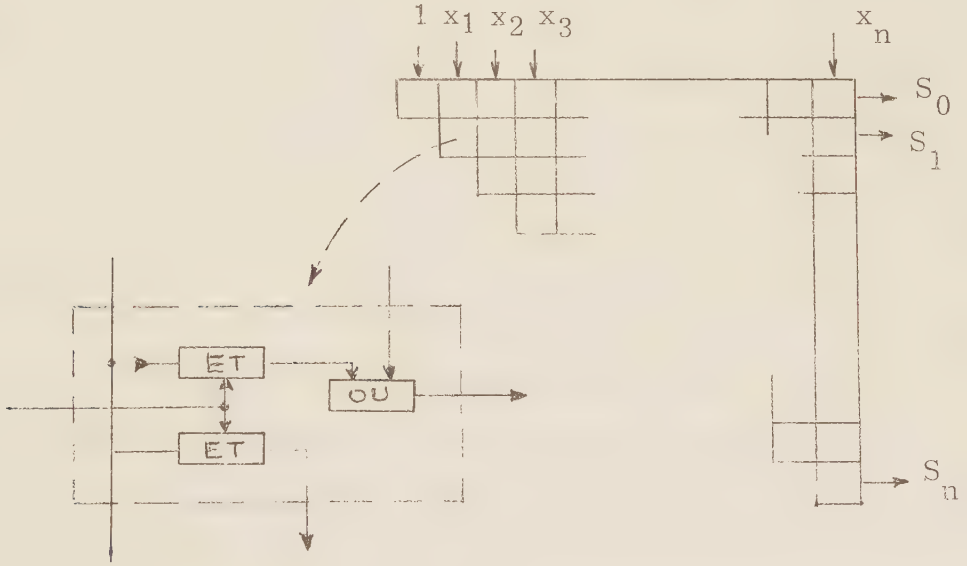


Figure I-6

Ce réseau est dérivé de la réalisation bien connue à relais de l'arbre des fonctions symétriques (CALDWELL). On peut écrire une fonction arbitraire f de ces variables exprimée sous forme d'un vecteur caractéristique $(c_{2^n-1}\ c_{2^n-2} \ldots c_j \ldots c_1\ c_0)$ comme une fonction symétrique de 2^n-1 variables

$$f = S_{a_1,\ a_2,\ \ldots,\ a_r}(x_1,\ x_2,\ x_2,\ x_3,\ x_3,\ x_3,\ \ldots,\ x_n)$$

où x_i se répète 2^{i-1} fois et les nombres a sont les indices j, un par chacun des digits f_j du vecteur caractéristique qui vaut 1. On peut voir aisément que les fonctions symétriques élémentaires dans cet arbre redondant sont les 2^n mintermes des variables $(x_1, x_2, \ldots x_n)$. Le réseau triangulaire résultant a les dimensions $2^n \times 2^n$.

I-2-1-3-4 - Réseaux cellulaires basés sur la décomposition d'une fonction logique sous forme de sommes de mintermes ou d'implicants premiers [6]

La façon la plus simple de construire un réseau de la forme décrite sur la figure I-5 est d'employer le développement canonique d'une fonction logique arbitraire f de n variables $f(x_1, x_2, \ldots x_n)$ sous forme de somme des mintermes

$$f = a_0\bar{x}_n \ldots \bar{x}_2\bar{x}_1 + a_1\bar{x}_n \ldots \bar{x}_2 x_1 + a_2\bar{x}_n \ldots x_2\bar{x}_1 + \ldots + a_{N-1}x_n \ldots x_2 x_1$$

L'ensemble des coefficients binaires $a_o, a_1, \ldots, a_{N-1}$, $N = 2^n$ spécifie complètement et de façon unique la fonction à réaliser. Pour une réalisation cellulaire chaque minterme vrai (c'est-à-dire pour lequel $a_i = 1$) pour une fonction donnée sera réalisé par une colonne distincte de cellules dans

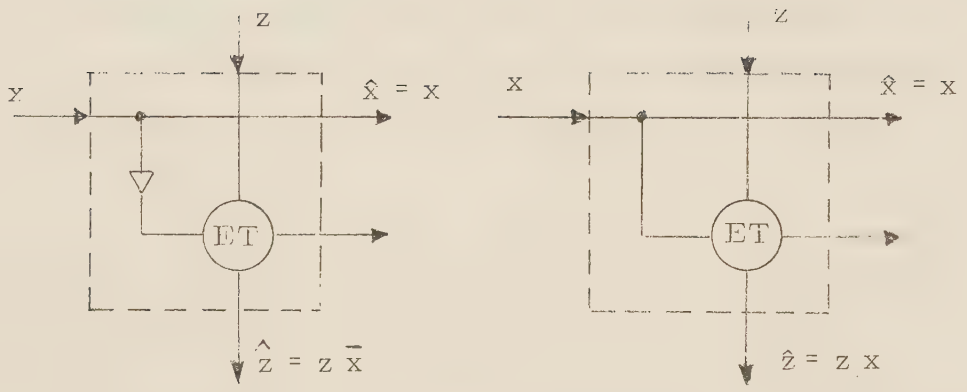
laquelle la cellule de la ligne j réalise dans la direction du bas le ET entre soit x_j, soit $\bar{x}_j$ et le produit partiel reçu de la cellule au-dessus. Désignant l'entrée et la sortie verticales d'une cellule respectivement par z et $\hat{z}$, et l'entrée et la sortie horizontales respectivement par x et $\hat{x}$, les équations des 2 modes de fonctionnement de la cellule peuvent s'écrire :

(1)
$$\begin{cases} \hat{z} = z\bar{x} \\ \hat{x} = x \end{cases} \quad \text{ou} \quad \begin{cases} \hat{z} = zx \\ \hat{x} = x \end{cases}$$

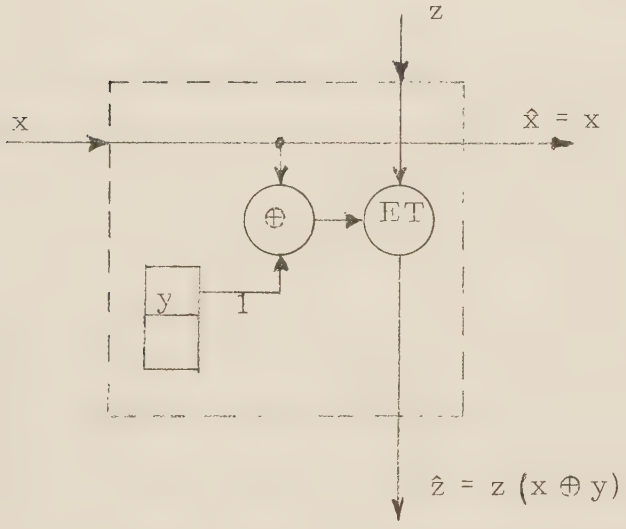
Ces deux modes de fonctionnement correspondent aux réalisations logiques présentées sur la figure I-7a. Indiquant le digit de programmation unique, qui est stockée dans une bascule, par y, la cellule complète est montrée sur la figure I-7b (excepté le circuit pour la remise à zéro et le positionnement de la bascule). Donc si les variables d'entrée $x_1, x_2, \ldots, x_n$ sont appliquées dans un ordre fixé sur le coté gauche d'un réseau (figure I-5) de cellules de ce type chaque colonne du réseau génère pour une programmation adéquate un minterme quelconque de ces variables.

La façon la plus simple de sommer un ensemble de ces mintermes, afin de former la fonction f est de combiner toutes les sorties inférieures des colonnes dans une porte OU extérieure au réseau cellulaire. Une alternative, spécialement appropriée quand on doit former simultanément plusieurs fonctions d'une même ensemble de variables d'entrée, est d'employer en plus du réseau initial un second réseau de sommation . Ce second réseau a la même structure d'interconnexion que le réseau initial (figure I-8).

Ce réseau doit permettre de faire la somme dans chacune de ses lignes d'un sous ensemble quelconque des entrées de ses colonnes.



(a)



(b)

Figure I-7 - Modes de fonctionnement et logique pour la cellule de base décrite par les équations (1)

a) les 2 modes de fonctionnement

b) la cellule complète

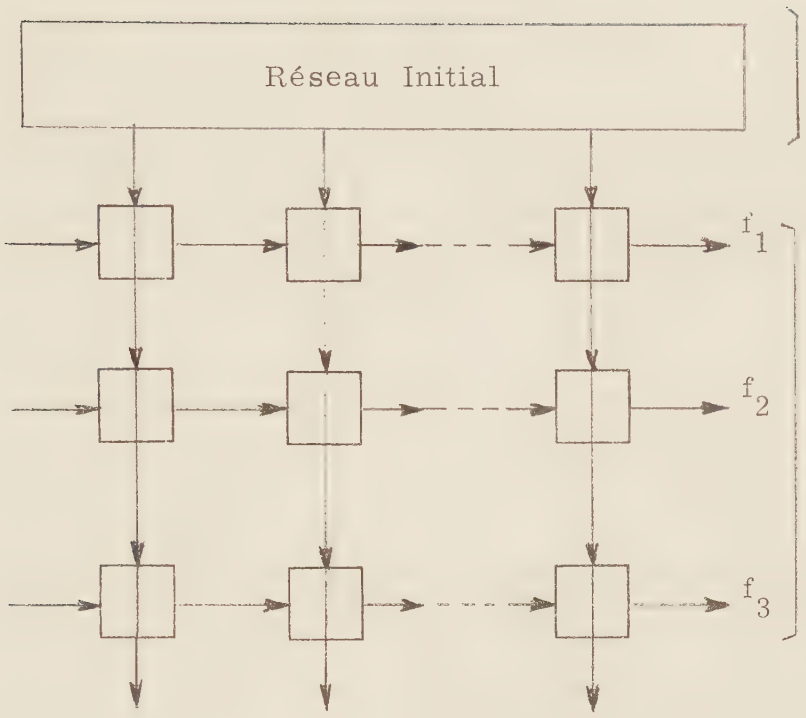
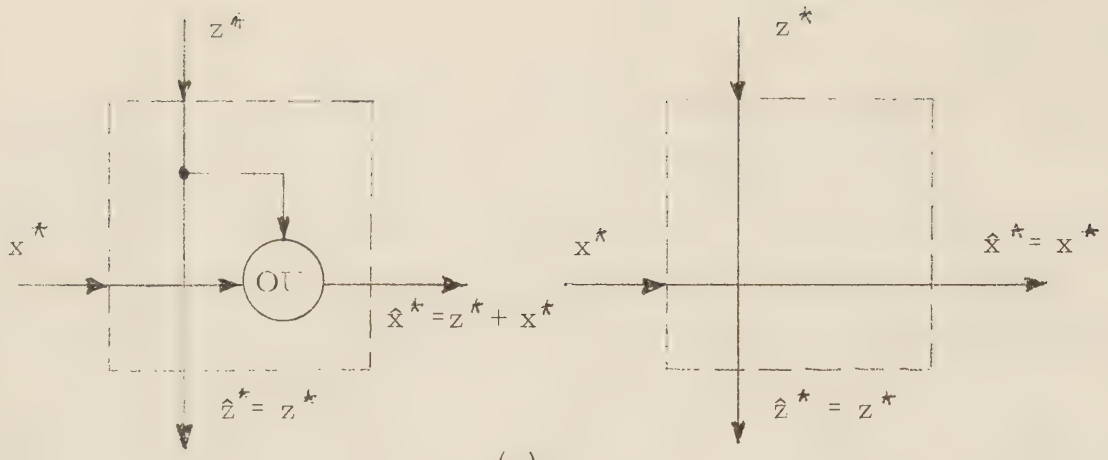


Figure I-8

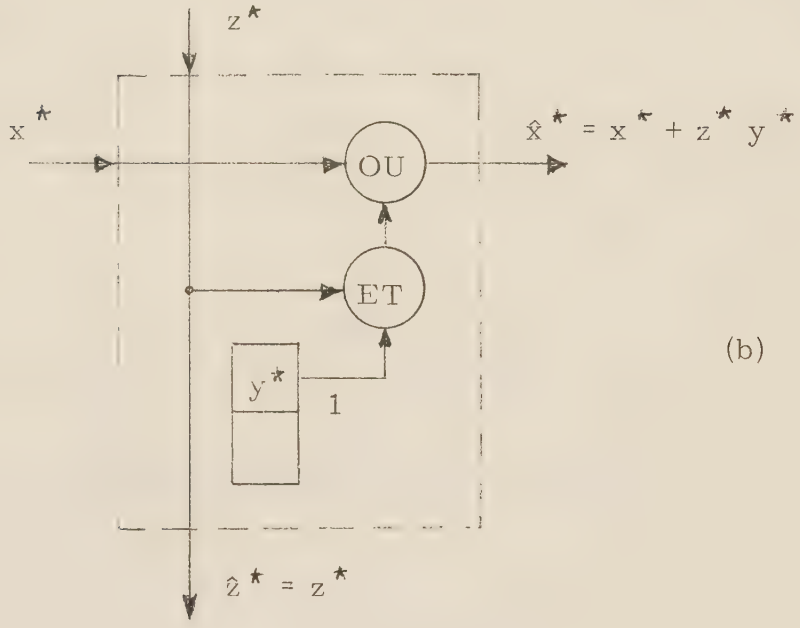
Donc chacune de ces cellules peut être décrite par les équations

$$(2)\quad \begin{cases} \hat{z}^{\star} = z^{\star} \\ \hat{x}^{\star} = x^{\star} \end{cases} \qquad \text{ou} \qquad \begin{cases} \hat{z}^{\star} = z \\ \hat{x}^{\star} = x^{\star} + z^{\star} \end{cases}$$

Ces deux modes de fonctionnement sont illustrés sur la figure I-9a et la cellule résultante sur la figure I-9b.



(a)



(b)

Figure I-9 - Modes de fonctionnement et logique pour la cellule de base correspondant aux équations (2),
a) les 2 modes de fonctionnement
b) la cellule

En conséquence m fonctions de n variables peuvent être réalisées par un réseau de cellules simples ayant n + m lignes et un nombre de colonnes égal au nombre le plus grand de mintermes requis par l'ensemble des m fonctions (2^n au plus).

Si nous désirons réaliser la fonction avec un seul type de réseau, nous pouvons fusionner les réseaux produisant les mintermes et le réseau de sommation. Sur la figure I-10 nous avons repliés le réseau de sommation et nous l'avons superposé au réseau de mintermes de sorte que chaque cellule contienne les circuits des 2 cellules constituantes. Elle possède deux bascules de programmation y et y^* qui génèrent quatre modes de fonctionnement et elle a deux connexions possibles dans chaque direction. Ses équations sont :

(3)

$\hat{z} = z\,\bar{x}$	$\hat{z} = z\,x$	$\hat{z} = z$	$\hat{z} = z$
ou	ou	ou	
$\hat{x} = x$	$\hat{x} = x$	$\hat{x} = x+z$	$\hat{x} = x$

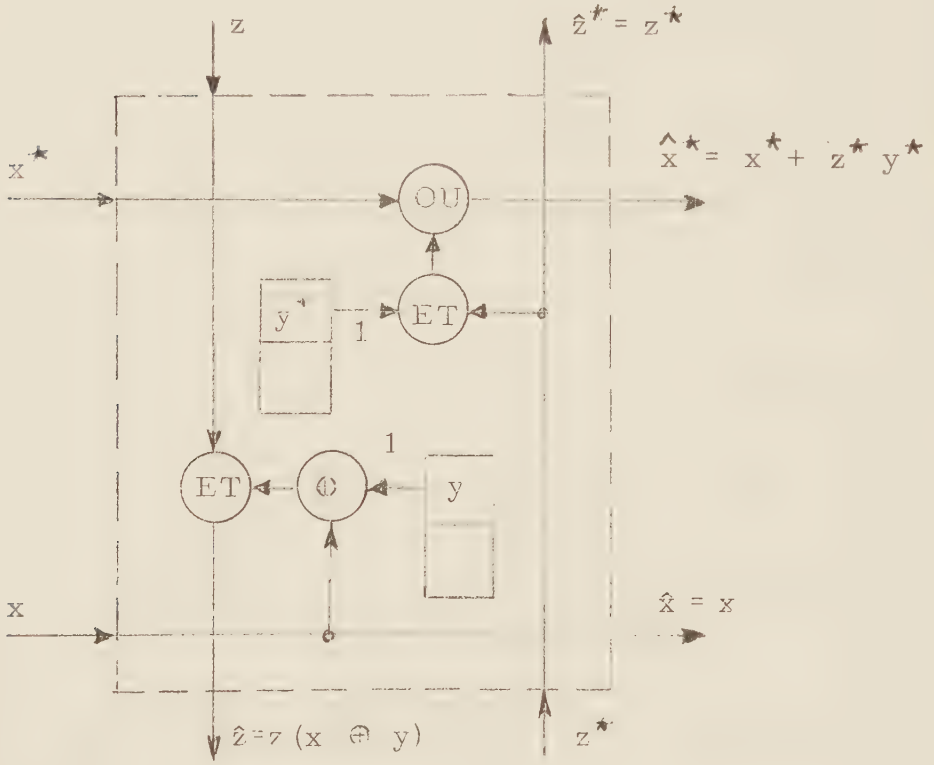


Figure I-10 - Combinaison des cellules des figures 8 et 9

I-2-1-3-5 - Réseaux cellulaires basés sur la décomposition galoisienne d'une fonction logique

Le développement galoisien d'une fonction logique arbitraire [13] est donné par :

$$(4) \qquad f = b_o \oplus b_1 x_1 \oplus b_2 x_2 \oplus b_3 x_1 x_2 \oplus b_4 x_3 \oplus \ldots \oplus b_{N-1}\, x_1 x_2 \cdots x_n$$

Par rapport au développement canonique, l'expression galoisienne emploie le OU EXCLUSIF au lieu du OU et chaque variable d'entrée complémentée x_i est remplacée par $1 \oplus x_i$. Ainsi, pour former le réseau correspondant les équations (1) de la cellule pour un réseau de mintermes sont remplacées par

$$(5) \qquad \begin{cases} \hat{z} = z \\ \hat{x} = x \end{cases} \quad \text{ou} \quad \begin{cases} \hat{z} = z\,x \\ \hat{x} = x \end{cases}$$

et les équations du réseau de sommation deviennent :

$$(6) \qquad \begin{cases} \hat{z}^* = z^* \\ \hat{x}^* = x^* \end{cases} \quad \text{ou} \quad \begin{cases} \hat{z}^* = z^* \\ \hat{x}^* = x^* \oplus z^* \end{cases}$$

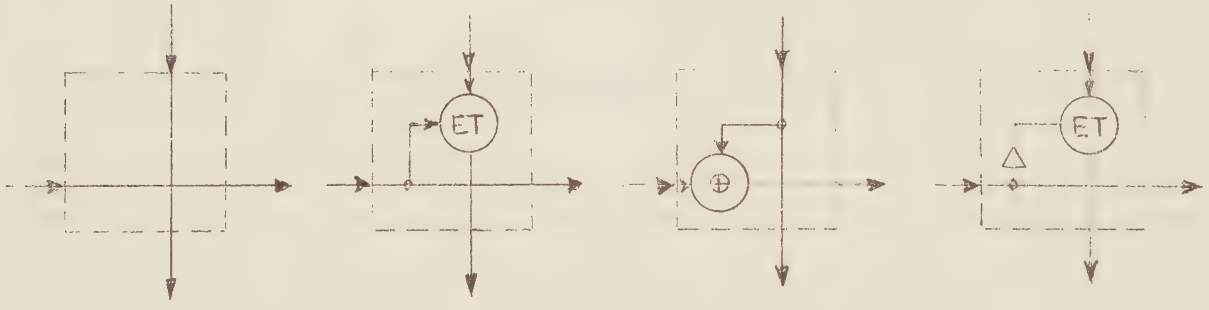
La combinaison des équations (5) et (6) pour arriver à un seul ensemble d'équations conduit à deux modes de fonctionnements identiques : les premiers modes de (5) et (6). Pour éviter cette duplication , nous pouvons tout aussi bien employer pour le quatrième mode une paire différente de fonctions logiques $\hat{z}$ et $\hat{x}$, choisie de façon à réduire au maximum la largeur du réseau par l'emploi d'un développement fonctionnel plus court.

Pour cela on peut employer pour le quatrième mode le premier mode des équations (3), ce qui conduit à :

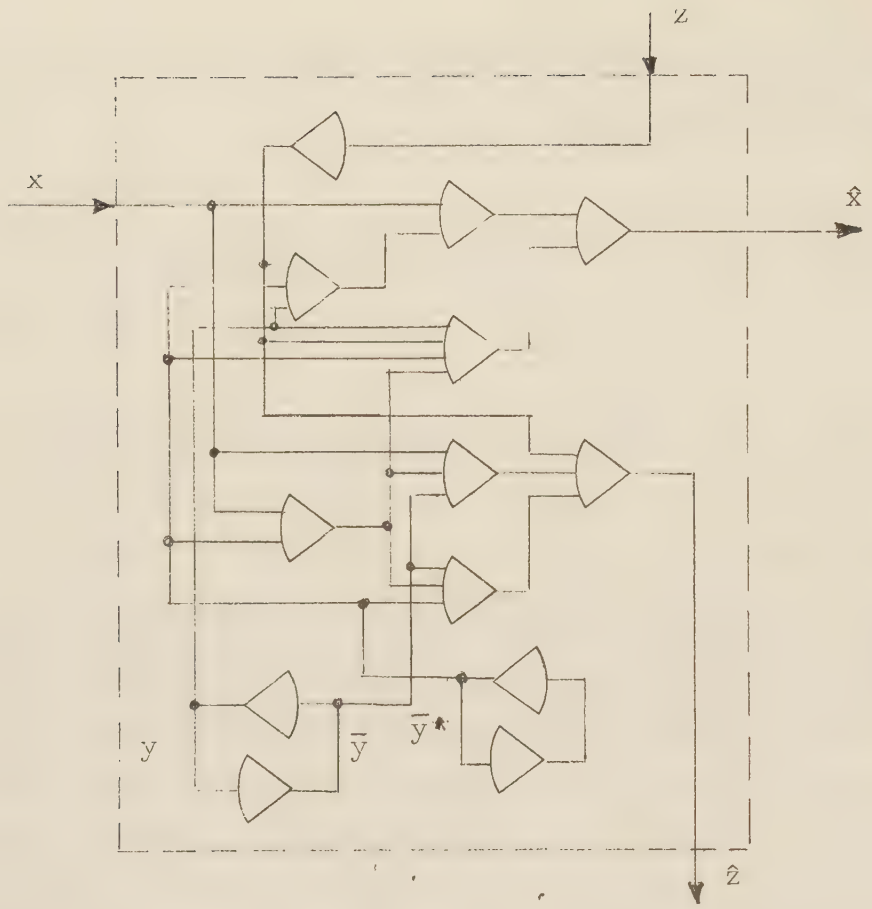
$$(7)\qquad \begin{array}{cccccccc} \hat{z} = z & & \hat{z} = zx & & \hat{z} = z & & \hat{z} = z\bar{x} \\ & \text{ou} & & \text{ou} & & \text{ou} & \\ \hat{x} = x & & \hat{x} = x & & \hat{x} = x \oplus z & & \hat{x} = x \end{array}$$

Les quatre modes de fonctionnement ainsi qu'une réalisation de la cellule complète avec des NI sont montrés respectivement sur les figure I-11 a et I-11b.

Dans ce cas le développement fonctionnel approprié est une somme OU EXCLUSIF de termes ayant l'aspect d'implicants premiers, c'est-à-dire que chaque terme est un simple produit des variables d'entrée choisies, vraies ou complémentées. Nous pouvons trouver facilement de courts développements de ce type pour des fonctions simples. Bien que des bornes sur la longueur de tels développements de fonctions quelconques soient connues, il n'existe encore aucun algorithme de synthèse efficace.



(a)



(b)

Figure I-11 - Modes de fonctionnement et logique pour la cellule complète
a) les quatre modes de fonctionnement
b) réalisation de la cellule en NI

I-2-1-3-6 - Réseaux de portes NI

Il est quelquefois plus pratique d'élaborer la réalisation cellulaire d'une fonction donnée directement à partir d'un réseau non cellulaire déjà existant formé de portes conventionnelles. Cette approche permet souvent à l'utilisateur de faire un plus grand usage de son expérience antérieure dans le domaine de la synthèse des circuits logiques. Si la réalisation initiale emploie de simples portes NI la cellule à quatre modes de fonctionnement décrite sur la figure I-12 est une façon de résoudre cette situation.

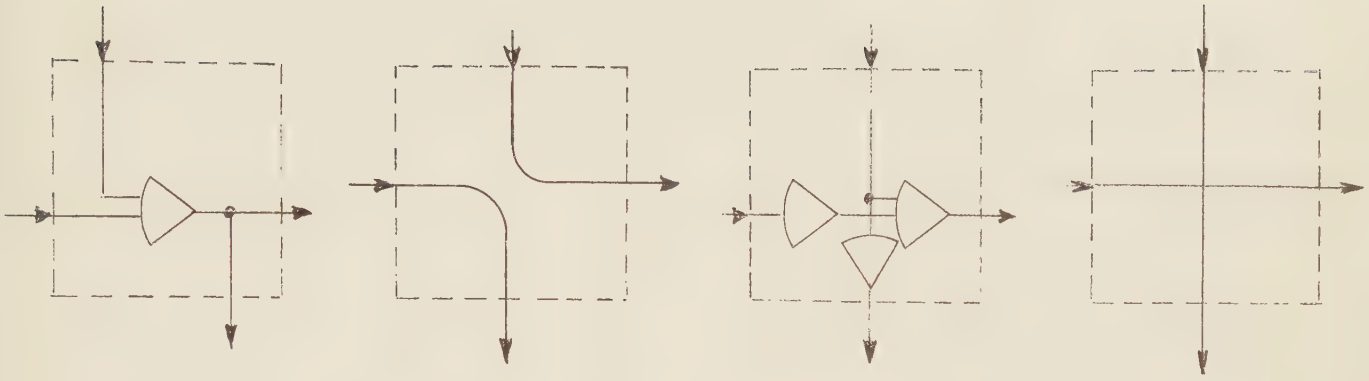


Figure I-12 - Les quatre modes de fonctionnement de la cellule

Les équations de la cellule sont :

$$(8)\quad \begin{array}{cccc} \hat{z} = \overline{zx} & \hat{z} = x & \hat{z} = \overline{z} & \hat{z} = z \\ \text{ou} & \text{ou} & \text{ou} & \\ \hat{x} = \overline{zx} & \hat{x} = z & \hat{x} = x\overline{z} & \hat{x} = x \end{array}$$

Les seconds et quatrième modes de cette cellule conduisent à une simple permutation. Le premier mode représente la porte NI elle-même distordue sous une forme cellulaire. Donc ces trois modes sont appropriés à la représentation sous forme cellulaire de tout réseau de portes NI à deux entrées sans boucles. La figure I-13 donne une représentation réelle de la cellule en opérateurs NI.

Cependant l'introduction d'un mode $\hat{z} = \overline{z}$, $\hat{x} = x\overline{z}$ permet une réduction substantielle du nombre total de cellules requis pour une réalisation et simplifie aussi considérablement la procédure d'éclatement du réseau donné en un réseau cellulaire.

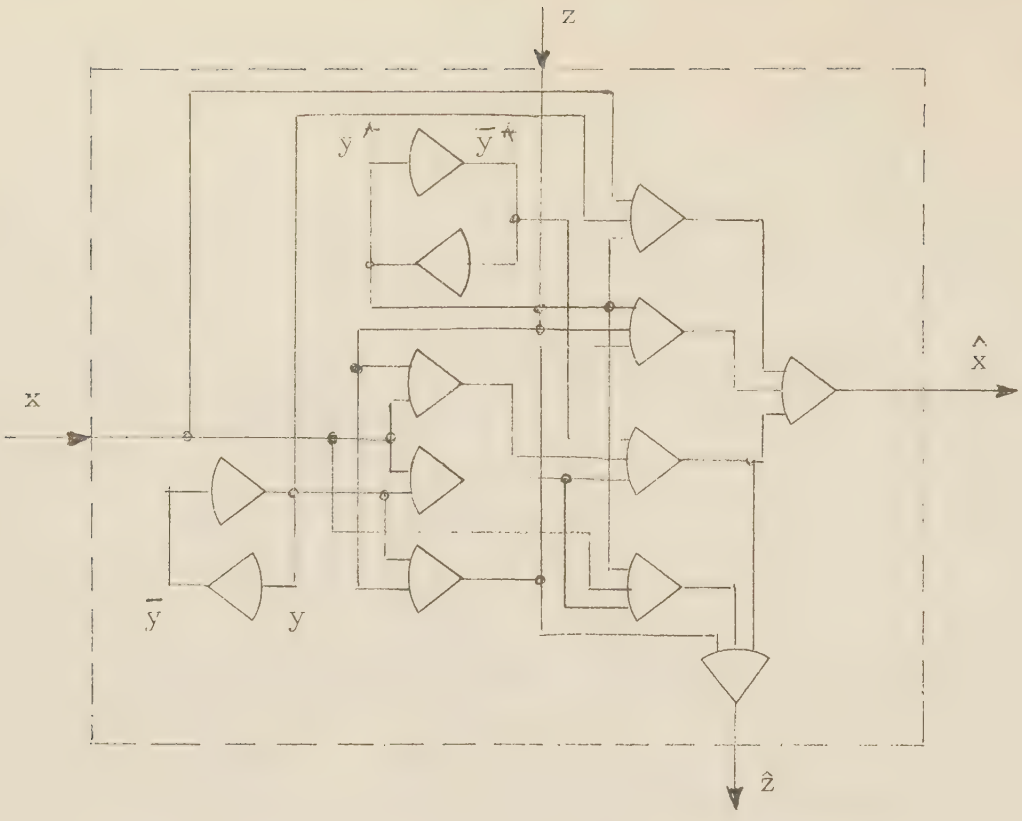


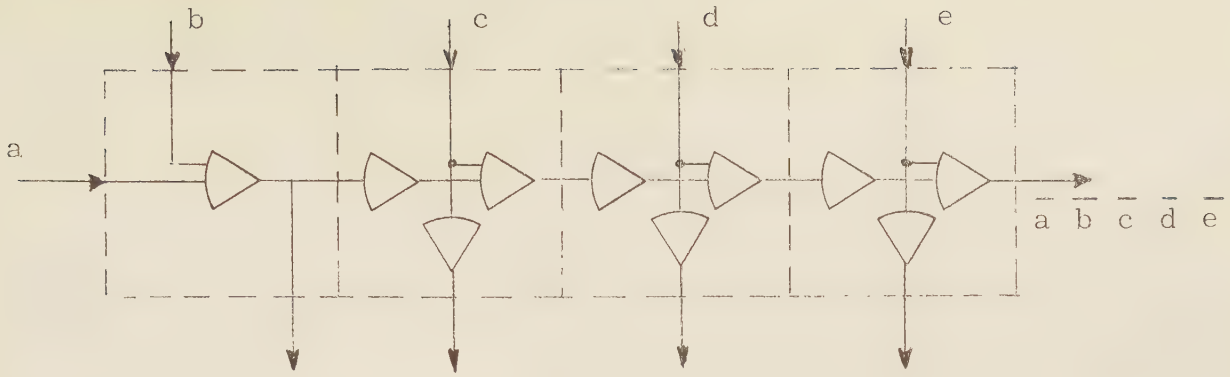
Figure I-13 - Réalisation de la cellule complète en NI

Les sous réseaux de la figure I-14 montrent la procédure de formation de sous fonctions importantes dans un réseau cellulaire. Notons en particulier l'emploi du 3ème mode pour former des produits de variables comme sur la figure I-14a,de sorte qu'une ligne de cellules se comporte comme une seule porte NI à entrance supérieure à 2.

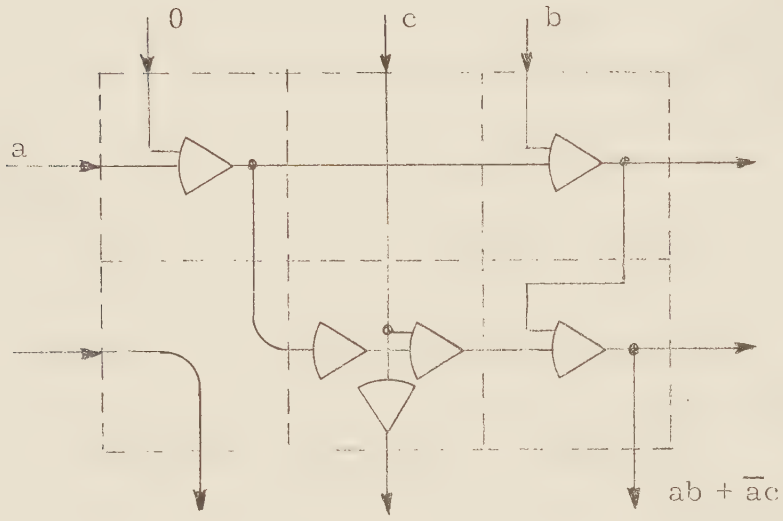
I-2-2 - Réseaux spécialisés

I-2-2-1 - Réseaux cellulaires pour la réalisation de fonctions à seuil [14]

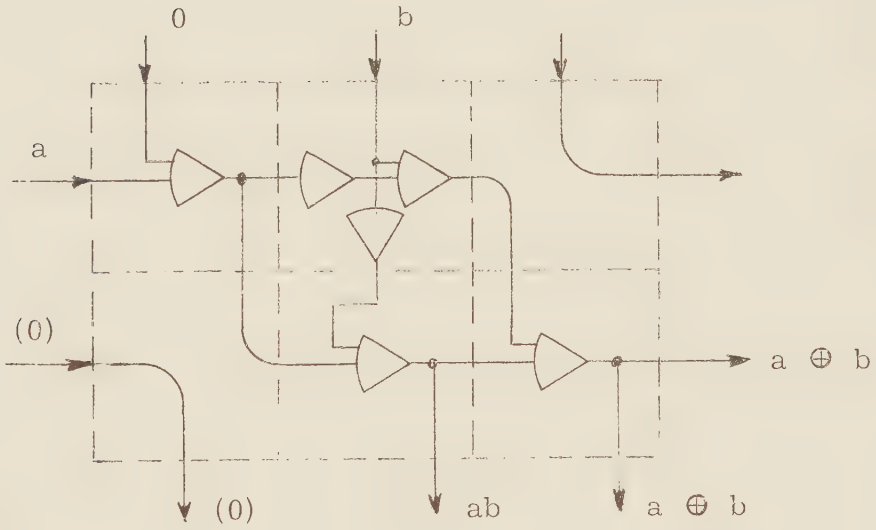
Une fonction logique à seuil $f(x_1, x_2, \ldots, x_n)$ des n variables d'entrée $x_1, x_2, \ldots, x_n$ est une fonction logique que l'on peut représenter sous la forme :



(a)



(b)



(c)

Figure I-14

a) NI

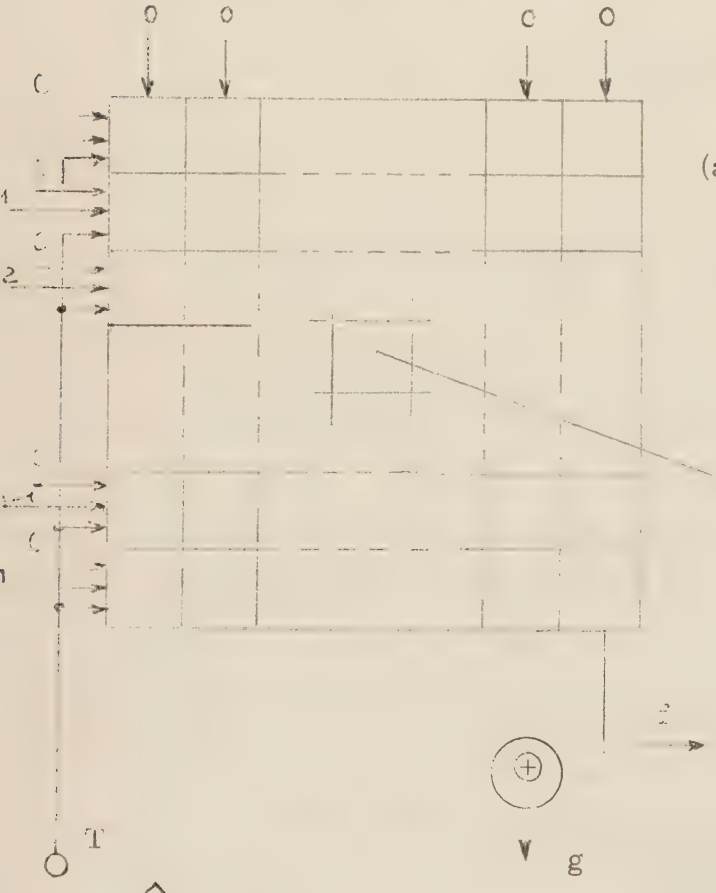
b) Commutateur

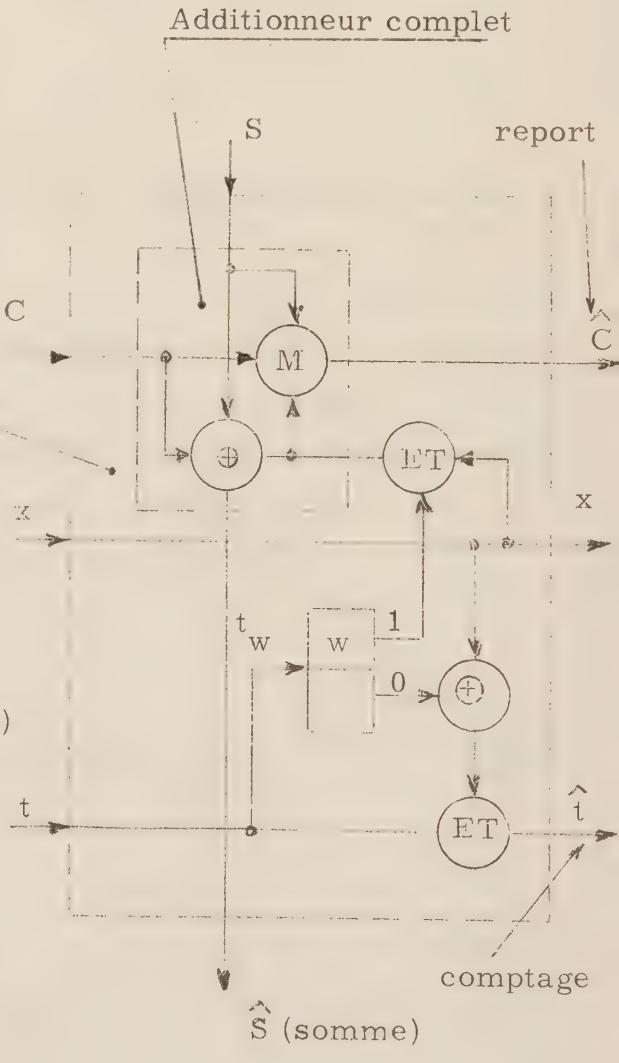
c) Demi additionneur

$$f = 1 \quad \text{si et seulement si} \quad \sum_{i=1}^{n} w_i x_i - w_0 \geq 0$$

où $\sum_{i=1}^{n}$ est une somme arithmétique, les w_i sont les poids (positifs ou négatifs) et l'entier w_0 est le seuil.

La figure I-15 illustre le réseau principal, une cellule de base, et les équations logiques de la cellule d'un arrangement cellulaire capable de réaliser toute fonction à seuil.





$\hat{S} = S \oplus C \oplus w\,x$

$\hat{C} = M\,(S,\ C,\ w\,x)$

$\hat{t} = t\,(\overline{w} \oplus x)$

$w' = w \oplus t$

(c)

Figure I-15 - Réseau cellulaire à seuil

a) le réseau principal

b) la cellule de base

c) les équations

Les variables d'entrée $x_1, x_2, \ldots, x_n$ sont distribuées par "bus" le long des lignes du réseau, une entrée par ligne. La fonction f est réalisée en sortie du coin le plus bas et à droite du réseau. Chaque cellule possède une bascule qui contient w, un additionneur binaire complet, avec des sorties $\hat{s}$ (somme) et $\hat{c}$ (report), et des entrées s (digit de somme de la cellule au dessus), c (digit de report de la cellule à gauche) et w x (contenu de la bascule et variable d'entrée x) ; et un circuit de comptage qui connecte l'ensemble des bascules w de chaque ligne comme un compteur décompteur que l'on décrira plus bas. Le réseau a n + 1 lignes et m colonnes.

Les m bascules de la $i^{ème}$ ligne de ce réseau réalisent la fonction d'un registre qui met en mémoire le $i^{ème}$ poids w_i, qui est un entier et est codé comme un nombre binaire à m digits (w_{i_m} , w_{i_2} w_{i_1}), le bit le moins significatif étant sur le côté gauche du réseau. Chaque poids négatif $w_i < 0$ est représenté sous sa forme complément à 2, c'est-à-dire un nombre binaire de valeur $2^m - w_i$. La ligne supérieure a gardé en mémoire le seuil sous la forme $(2^{m-1} - w_0)$. L'ensemble des n additionneurs complets de la $i^{ème}$ ligne du réseau forme une additionneur complet parallèle qui ajoute le poids stocké w_i, dont on fait le ET avec la valeur de x_i, au nombre reçu en parallèle de la ligne au-dessus ; cette somme est transmise à la ligne en dessous. Donc le nombre apparaissant au bas du réseau est

$$S \equiv \sum_{i=1}^{n} w_i x_i - w_0 + 2^{m-1} \pmod{2^m}$$

Pourvu seulement que m soit assez grand pour prendre en compte les excursions extrèmes de la valeur de S, quand les entrées x_i varient, cette somme S sera plus grande ou plus petite que 2^{m-1} selon que la portion de la somme faisant intervenir les poids est plus grande ou plus petite que zéro, c'est-à-dire S tombera dans l'intervalle $2^{m-1} \leqslant S \leqslant 2^{m-1}$ si et seulement

si $\sum_{i=1}^{n} w_i x_i - w_0 \geq 0$ qui représente l'inégalité définissant la condition de sortie f = 1 pour un élément à seuil. Cette condition sur l'intervalle de variation de S est contenue dans la valeur de son bit le plus significatif, qui vaut 1 si et seulement si S est au moins aussi grand que 2^{m-1}. Donc $f = s_{mn}$.

Donc un réseau ayant en mémoire les poids $w_1, w_2, \ldots, w_n$ (et la valeur du seuil $2^{m-1} - w_0$ dans la ligne zéro) et ayant les connections d'entrée et de sortie montrées sur la figure I-15 réalise la fonction à seuil donnée par l'inégalité ci-dessus. Le plus long chemin de propagation à travers le réseau comprend n + m portes. Ce réseau a un total de n + 1 lignes, une pour chacune des variables d'entrée, plus une pour le seuil. Le nombre m de colonnes doit être assez grand pour tenir compte à la fois des valeurs extrêmes négative et positive de la somme commutative des poids, qui peut se produire dans toute ligne quand on permet aux variables d'entrée de prendre toutes leurs valeurs possibles. Ces valeurs extrêmes sont atteintes au bas du réseau, de sorte que m doit satisfaire les 2 inégalités.

(9) $\quad 2^{m-1} \geq \sum_{+} w_i - w_0 \qquad 2^{m-1} \geq \sum_{-} |w_i| + w_0$

$\sum_{+}$ et $\sum_{-}$ prises respectivement sur les poids positifs et négatifs.

Exemples : réalisation de la fonction $f_1(x_1, x_2, x_3)$ définie par

$f_1 = 1$ si et seulement si $3x_1 - 2x_2 + x_3 \geq 2$

(9) donne

$2^{m-1} \geq 3 + 1 - 2 = 2$ et $2^{m-1} \geq 2 + 2 = 4$ aussi $m = 3$

donc les bascules des lignes 0, 1, 2 et 3 doivent stocker respectivement les nombres binaires $2^2 - w_0 = 2 = (0\ 1\ 0)$, $w_1 = 3 = (0\ 1\ 1)$, $w_2 = 2 = (1\ 1\ 0)$

et $w_3 = 1 = (0\ 0\ 1)$ ce qui est représenté sur la figure I-16.

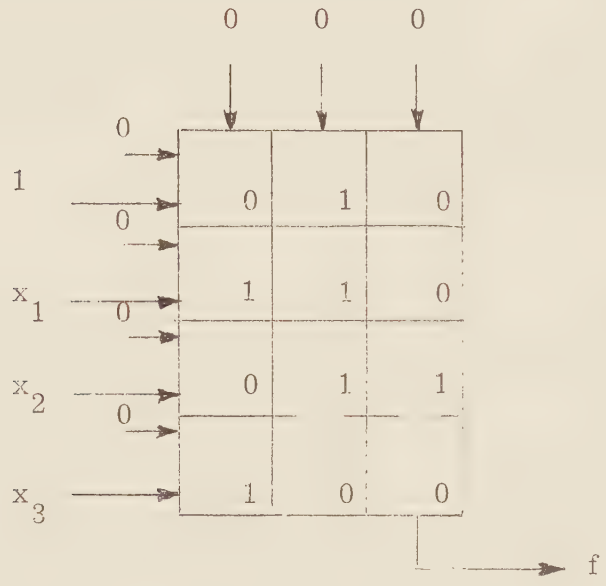


Figure I-16 - Contenus des bascules des cellules pour l'exemple traité

Comme on peut le voir sur la figure I-15 les m bascules de la première ligne du réseau sont cablées comme un compteur décompteur avec une entrée commune T. Quand T est appliqué les compteurs de toutes les lignes du réseau sont changés en même temps : le compte augmente de 1 dans chaque ligne pour laquelle $x_i = 1$ et décroît de 1 dans chaque ligne pour laquelle T = 0.

I-2-2-2 - Réseaux cellulaires d'interconnexion [15]

Une des fonctions fondamentales des processeurs de données est la commutation de données. Nous décrivons ici un exemple de l'utilisation que peuvent avoir les réseaux cellulaires dans ce domaine.

La figure I-17 représente une cellule simple capable de réaliser une opération de commutation de ligne. C'est un commutateur à double pôle et double sens commandé par un élément mémoire (bascule), les deux modes de fonctionnement sont illustrés sur la figure I-17 qui donne aussi une réalisation de la cellule à l'aide d'opérateurs NI.

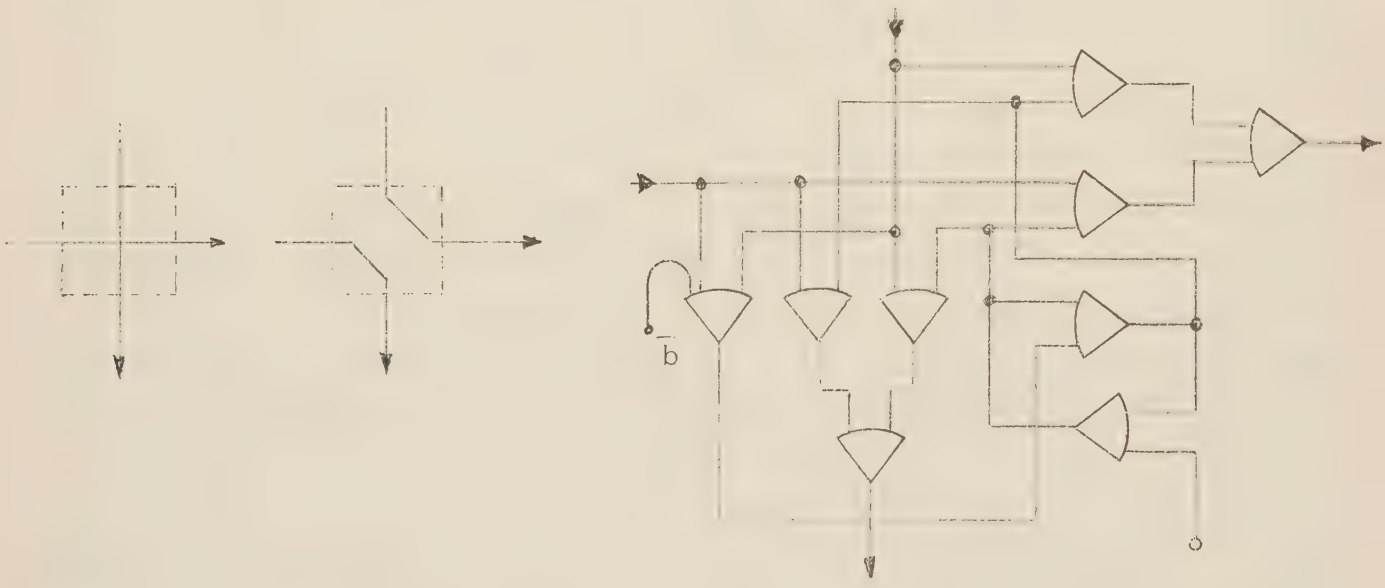


Figure I-17 - Modes de fonctionnement et réalisation de la cellule de base

La cellule par elle même permet un simple échange de deux lignes et constitue une solution pour n = 2 au problème général de la synthèse d'un réseau d'interconnexion à n entrées et n sorties.

On peut obtenir aisément une borne inférieure du nombre M (n) de cellule élémentaires reprises par un réseau d'interconnexion à n sorties et n entrées : le réseau doit contenir assez d'éléments mémoire (cellules) pour spécifier toutes les permutations possibles ; donc $M(n) \geq \log_2 (n\ )$

La figure I-18 représente pour n = 8 la forme d'un réseau d'interconnexions possible.

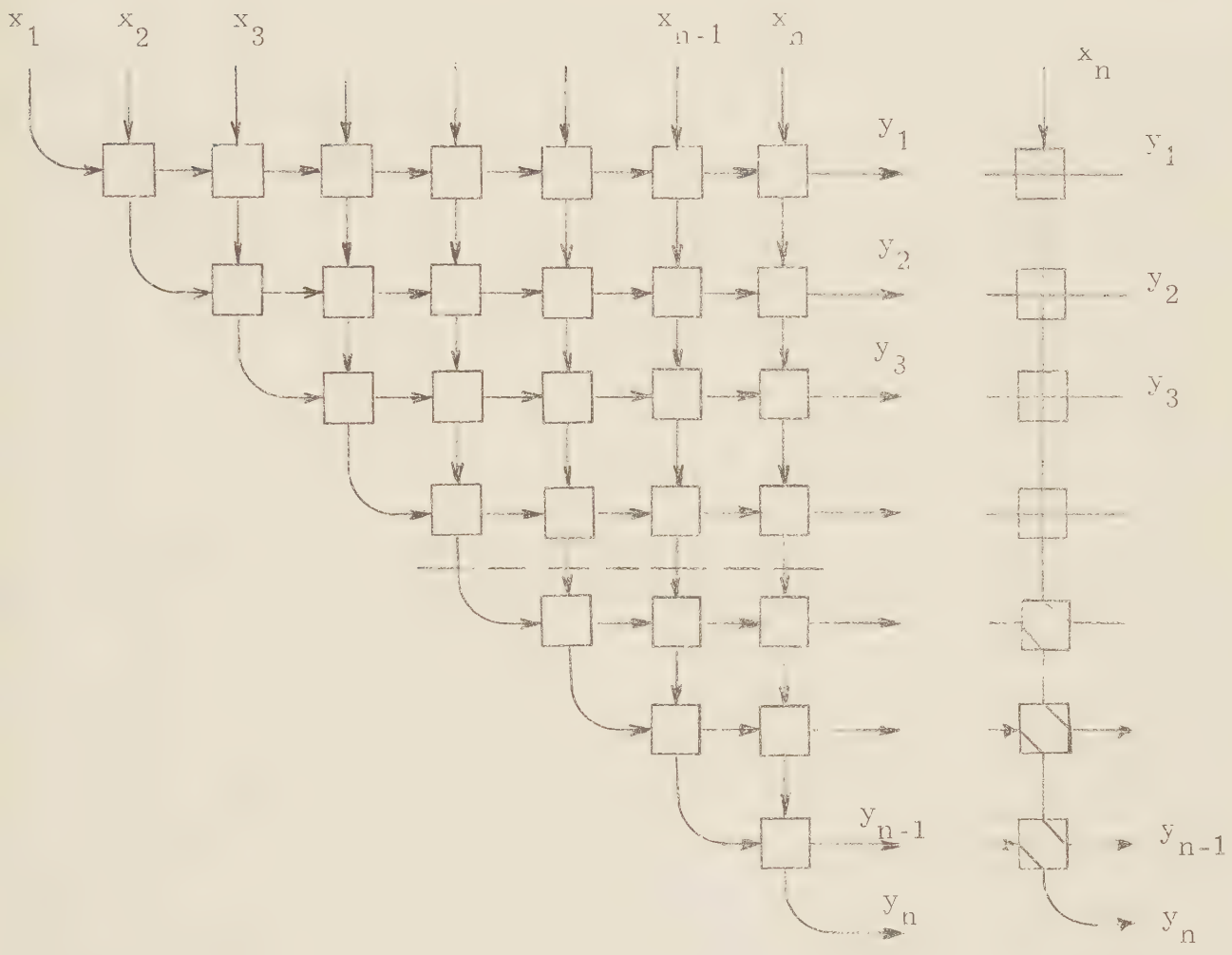


Figure I-18 - Réseau d'interconnexion

a) le réseau

b) la dernière colonne

On peut prouver aisément par induction son aptitude à réaliser une permutation arbitraire des n entrées. La cellule la plus à gauche réalise d'elle même la permutation pour n = 2. Supposons que les n - 2 colonnes les plus à gauche (toutes exceptée la dernière sur la figure) soient capables de permuter l'ensemble des lignes d'entrée $x_1, x_2, \ldots, x_n$) dans un ordre désiré arbitrairement choisi à la droite de la n - $2^{ème}$ colonne. Pour obtenir la permutation finale des n variables d'entrée il est nécessaire et suffisant de positionner une partie des cellules de la dernière colonne dans leur premier mode de fonctionnement et l'autre partie dans leur deuxième mode. Par induction alors, un total de (n-1) colonnes est approprié pour réaliser une permutation arbitraire des n lignes d'entrée.

Le nombre total de cellules dans ce réseau est :

$$M(n) = 1 + 2 + \ldots + (n - 1) = \frac{1}{2} \quad (n^2 - n)$$

Afin de programmer un réseau d'interconnexion pour réaliser une permutation donnée, on applique une séquence de signaux d'entrée binaires aux entrées du réseau de façon à insérer l'échantillonnage désiré de digits binaires dans toutes les bascules des cellules. On peut employer des lignes "bus" spéciales pour programmer le réseau d'une façon directe. Pour le type de cellules de la figure I-17 cependant, chaque bascule est positionnée par une coïncidence de signaux 0 sur ses entrées horizontales et verticales en simultanéité avec une impulsion d'horloge appliquée globalement. On suppose que toutes les bascules des cellules peuvent être initialement remises à 0, de sorte que chaque cellule fonctionne dans son premier mode au départ de la procédure.

On programme le réseau, une cellule à la fois, en appliquant à la frontière du réseau à chaque pas deux 0, un qui doit parvenir à l'entrée horizontale et l'autre à l'entrée verticale de la cellule en question. De tels chemins existent à coup sur à cause du fait que chaque cellule permute simplement ses entrées, de sorte que les chemins de propagation ne peuvent

ni se terminer ni se combiner. Cependant les cellules doivent être positionnées dans un ordre tel que les chemins transportent les signaux 0 à travers le réseau ne s'intersectent que dans la cellule dont on est entrain de positionner le flip flop.

Pour le réseau diagonal décrit figure I-19, les diagonales doivent être positionnées successivement, de la plus longue à la plus courte. Les cellules de chaque diagonale peuvent être positionnées dans n'importe quel ordre. On peut voir sur l'exemple de la figure I-19 que pour chaque cellule de la diagonale qu'on est entrain de programmer (la troisième diagonale), un des signaux 0 incidents vient directement d'au-dessus, tandis que l'autre vient de plus loin à gauche. De la même façon un des signaux 0 sortant va directement à droite, tandis que l'autre suit un chemin plus bas vers le côté droit. Donc l'intersection de deux signaux 0 ne peut se produire excepté dans la cellule en question. Un argument identique s'applique à toutes les cellules d'une diagonale de sorte que l'on peut programmer le réseau entier selon n'importe quel échantillonnage désiré.

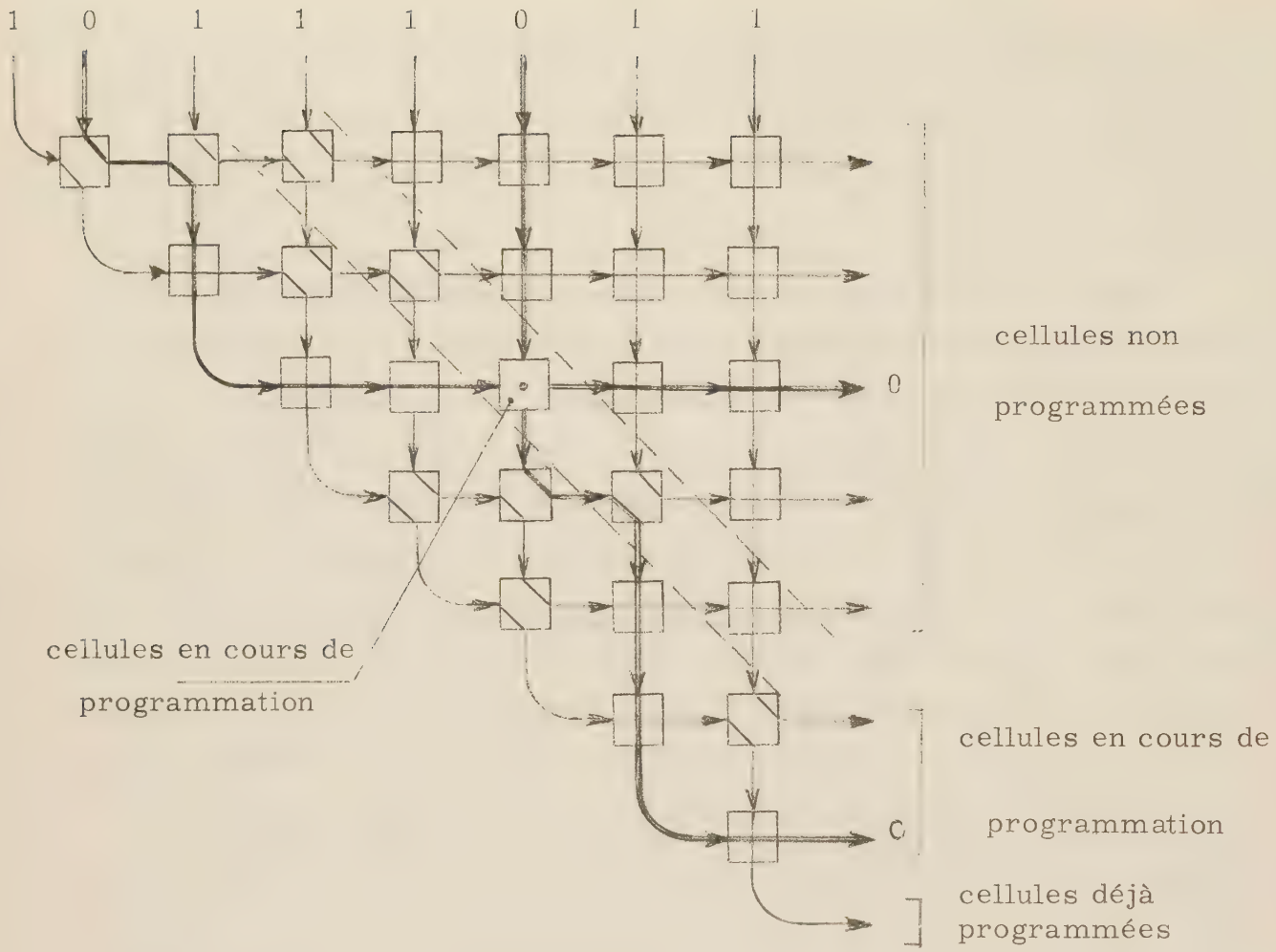


Figure I-19

I-3 - MINIMISATION DES RESEAUX CELLULAIRES

Pour les réalisateurs logiques, un des problèmes fondamentaux de la logique cellulaire est la construction de réseaux cellulaires qui optimisent un ensemble de paramètres. Les paramètres concernés sont nombreux et compliqués ; une liste partielle de ceux-ci pourrait être : les dimensions de réseau mesurées par le nombre de cellules, la complexité logique de la cellule ainsi que le nombre de bits de spécification ou, la structure d'interconnexion, le nombre total de connexions internes et externes de la cellule, la vitesse de propagation du signal qui dépend de la distance maximum entre cellules, et

nombre d'autresparamètres électriques et géométriques propres à une technologie donnée. Certains chercheurs ont donné des résultats analytiques limités sur l'interdépendance entre quelques uns des paramètres purement topologiques tels que nombre total de connexions incidentes à une cellule, distance maximum et moyenne entre cellules, dimensions et planarités des réseaux. Quand nous considérons l'ensemble des paramètres logiques et topologiques, la situation devient extrèmement difficile à définir mathématiquement. De plus un ensemble de définitions adoptées aujourd'hui peut devenir démodé demain par suite de l'évolution de la technologie. Même si nous adoptons un ensemble de définitions, il est mathématiquement difficile d'arriver à des solutions minimales dans les cas les plus généraux.

Certains auteurs [16, 17] ont développé des algorithmes de minimisation pour des réseaux particuliers. Nous donnons ci-dessous un bref aperçu de la méthode présentée par A. MUKHOPADHYAY pour les cascades cellulaires monotones qui sont des cascades de Maitra dans lesquelles la cellule de base ne réalise que les quatorze fonctions monotones de ses deux entrées. Les propriétés suivantes de ces cascades peuvent être prouvées aisément (i) la sortie d'une cascade monotone est une fonction monotone ; (ii) chaque cascade monotone a une forme canonique irrédondante dans laquelle chaque cellule est soit une fonction ET soit une fonction OU ; (iii) une cascade monotone irrédondante dans laquelle chaque cellule ne produit que les fonctions (1), (2), (3), (5), et (6) du théorème I peut réaliser toutes les fonctions d'une cascade quelconque ; (iv) la forme somme de produits d'une fonction monotone (sortie d'une cascade monotone) donne l'ensemble des implicants premiers essentiels de la fonction. Le problème de la minimisation pour une cascade monotone réalisant une fonction f est le suivant : trouver un ensemble de fonctions monotones u_1, u_2, ..., u_k telles que

(a) soit $f = \sum_{i=1}^{k} u_i$ soit $f = \prod_{i=1}^{k} u_i$

(b) chacune des fonctions monotones u_i est réalisable par une cascade monotone ayant les mêmes variables d'entrée

(c) k est minimum.

Le développement de réseaux basés sur des expressions contenant le OU EXCLUSIF ou l'EQUIVALENCE comme opérateurs de base pose aussi un problème de minimisation : trouver une procédure systématique pour exprimer une fonction donnée arbitraire sous la forme d'une somme OU EXCLUSIF ou d'une somme EQUIVALENCE d'un nombre minimum de fonctions réalisables par une cascade. Ce problème difficile n'a pas encore reçu de solutions.

Des problèmes de minimisation qui concernent les paramètres purement topologiques du réseau ont aussi été étudiés et n'ont pas conduit à des solutions satisfaisantes.

Nous pensons que la minimisation des réseaux cellulaires reste un problème ouvert qui nécessitera encore de nombreuses études.

I-4 - CONCLUSION

Nous avons décrit dans ce chapitre un certain nombre de réseaux qui permettent d'évaluer l'intérêt des méthodes et des techniques de la logique cellulaire. Nous pensons avoir ainsi fait apparaître quelques uns des problèmes posés par le développement de cette technique : choix de la cellule de base, choix de la structure d'interconnexion, algorithmes de synthèse, etc, pour lesquels nous essaierons d'apporter une contribution dans les chapitres ultérieurs.

CHAPITRE II

ANALYSE ET SYNTHESE DE CIRCUITS CELLULAIRES PARTICULIERS A PARTIR DE METHODES BOOLEENNES CLASSIQUES

II-1 - ANALYSE ET SYNTHESE DE CASCADES DE MAITRA

Dans ce paragraphe nous décrirons et étudierons des structures cellulaires du type "Maitra" dont les cellules de base sont composées d'opérateurs NI ON. Nous avons présenté très rapidement dans le chapitre I quelques unes des caractéristiques essentielles de ces cascades dont un exemple est donné sur la figure II-1.

Figure II-1 - Cascade de Maitra

II-1-1 - Rappels sur les cascades de Maitra [7]

Maitra a proposé en 1962 une cellule à deux entrées et une sortie pouvant réaliser les seize fonctions de ces deux variables d'entrée. Il a montré l'intérêt que présentait ce type de cellules dans un réseau en cascade et donné une méthode d'énumération et de synthèse des fonctions réalisables par ces réseaux. Cette méthode basée sur une construction graphique s'avère très difficilement exploitable pour des fonctions d'un grand nombre de variables.

Un des résultats importants de ces études concerne le nombre de fonctions réalisables par ces structures qui décroît rapidement lorsque le nombre de variables augmente. Ce résultats négatif ne suffit cependant pas à rendre inintéressantes ces cascades qui, du fait de leur structure itérative, du nombre limité d'interconnexions et de leur simplicité se prêtent aisément à la techno-

logie de fabrication en circuits intégrés.

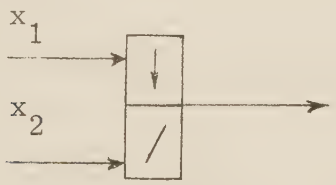
Nous décrirons ici l'approche que nous préconisons pour l'étude de cascades de Maitra irrédondantes et non permutables. Dans une première étape nous nous arrêterons quelques instants sur l'élaboration d'une cellule de base particulière.

II-1-2 - Définition et propriétés d'une cellule de base composée d'opérateurs NI-ON [18, 19, 20]

II-1-2-1 - Propriétés des opérateurs NI-ON

Définition : Un opérateur NI-ON est un opérateur logique complet qui peut réaliser soit la fonction NI soit la fonction ON.

Notation : Un opérateur NI-ON sera représenté par le symbole de la figure II-2.



$x_1 \downarrow x_2$ ou x_1 / x_2

Remarques : $x_1 \downarrow x_2 = \overline{x}_1 \cdot \overline{x}_2$

$x_1 / x_2 = \overline{x}_1 + \overline{x}_2$

Figure II-2 - Opérateur NI-ON

Le choix de cet opérateur nous a été suggéré par des recherches antérieures que nous avions effectuées sur les réseaux arborescents composés d'opérateurs NI-ON. Nous avions donné alors une méthode de synthèse gra-

phique des fonctions logiques à l'aide de ces opérateurs [18] . Cette méthode est systématique et se prête aisément à la programmation qui est indispensable dans le cas d'un nombre de variables élevé. De plus elle permet une réduction du coût par rapport aux méthodes de synthèse à l'aide d'opérateurs NI ou ON (Le coût étant défini comme le nombre de modules nécessaires à la réalisation de la fonction booléenne).

II-1-2-2 - Réalisations électroniques d'un opérateur NI-ON

Nous présentons ici sans les détailler deux réalisations électroniques possibles pour un opérateur NI-ON à trois entrées (figures II-3 et II-4). Ces circuits ont été étudiés en collaboration avec des chercheurs du groupe "Composants" du L. A. A. S. Des discussions communes ont permis d'aboutir à la définition de deux schémas électroniques utilisant comme élément de base le transistor M. O. S. L'intégration de ces circuits ne présente pas de difficultés majeures et le coût de réalisation bien que plus élevé que pour des circuits NI ou ON classiques reste dans des limites raisonnables surtout dans la perspective d'une grande série.

Dans les deux montages les signaux d'entrée A, B et C sont appliqués sur les grilles des transistors M. O. S. dont les résistances de charge sont aussi des transistors M. O. S. Le passage de la fonction NI à la fonction ON s'effectue par un bit de programmation S qui prend la valeur 0 ou 1 selon que la fonction réalisée par le module est ON ou NI.

II-1-2-3 - Cellule de base de la cascade

Les recherches effectuées sur les propriétés des opérateurs NI-ON nous ont permis de définir une cellule de base plus complexe, utilisable en particulier dans les réseaux cellulaires en cascade. Cette cellule doit posséder les caractéristiques suivantes :

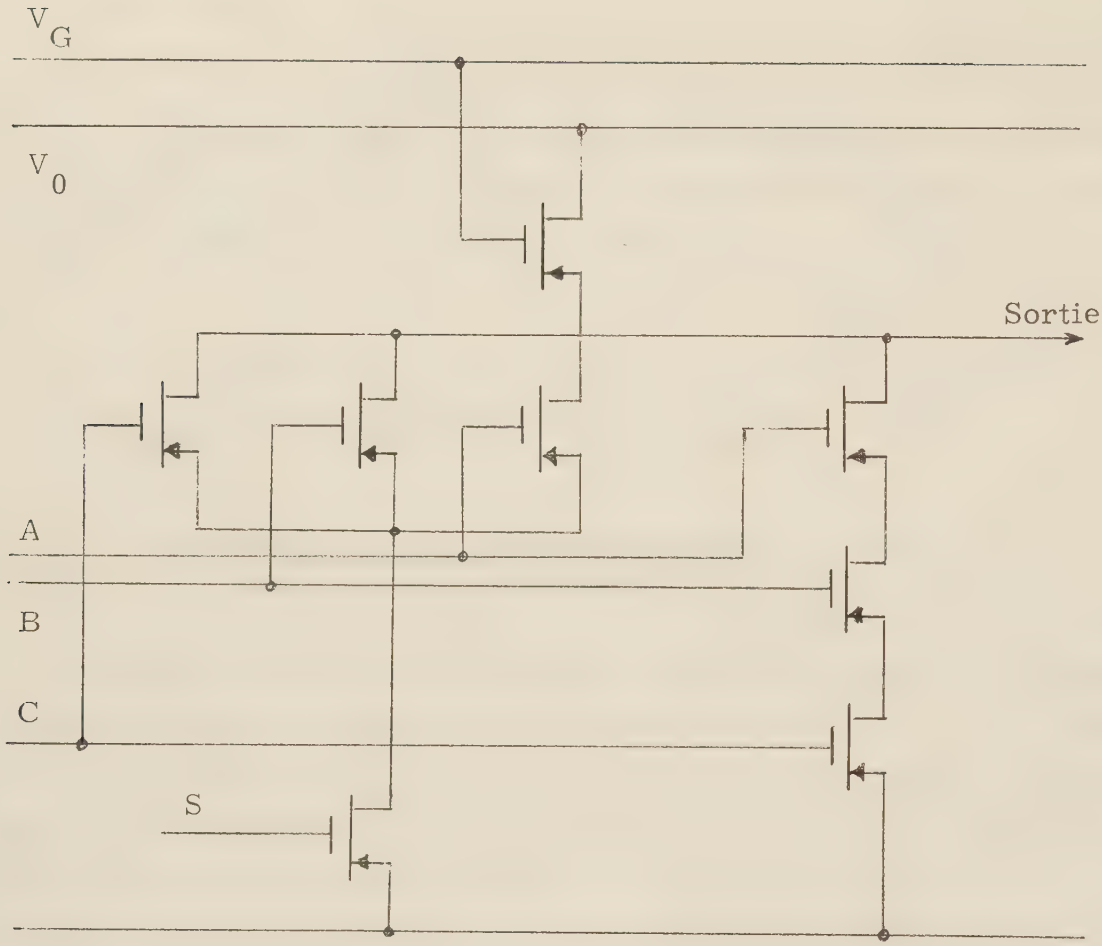


Figure II-3 - Première réalisation de l'opérateur NI-ON

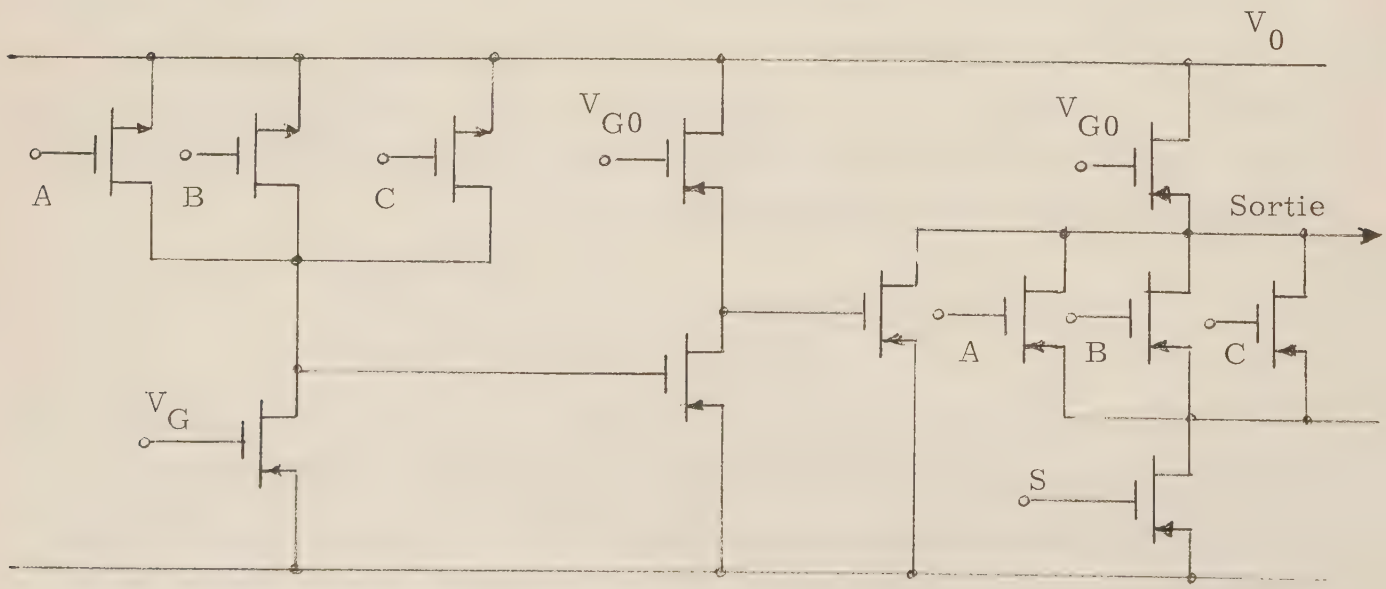


Figure II-4 - Deuxième réalisation de l'opérateur NI-ON

- deux entrées et une sortie
- réaliser les seize fonctions de ses deux variables d'entrée
- posséder une structure d'interconnexion interne simple et régulière permettant une intégration peu coûteuse
- comporter un nombre minimum d'opérateurs NI-ON
- avoir un nombre de couches minimum.

Ces différentes caractéristiques nous ont amené à choisir une cellule de base ayant la structure représentée sur la figure II-5.

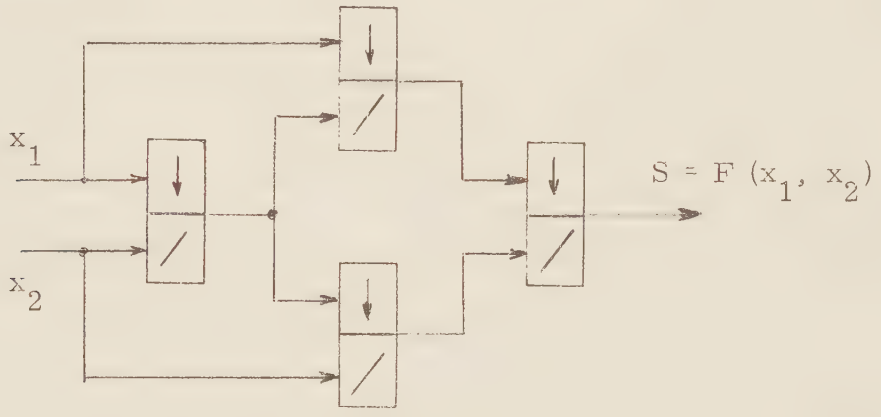


Figure II-5

La cellule comporte quatre opérateurs NI-ON. Les signaux d'entrée sont appliqués en x_1 et x_2. Il est aisé de voir que la sortie S réalise les seize fonctions des deux variables d'entrée x_1 et x_2 à condition d'admettre comme signaux d'entrée x_1, $\bar{x}_1$, x_2, $\bar{x}_2$ et les constantes 0 et 1. De plus cette cellule nécessite quatre entrées supplémentaires comme digits de commande. Ces digits permettent d'afficher chaque cellule soit à la fonction NI soit à la fonction ON.

Remarques

1. Pour des vitesses de fonctionnement élevées cette cellule présente des aléas de propagation. En effet le nombre de modules commandés par les signaux d'entrée x_1 et x_2 est différent suivant les trajets considérés. Ce problème peut être résolu par l'adjonction de deux opérateurs NI-ON supplémentaires (figure II-6).

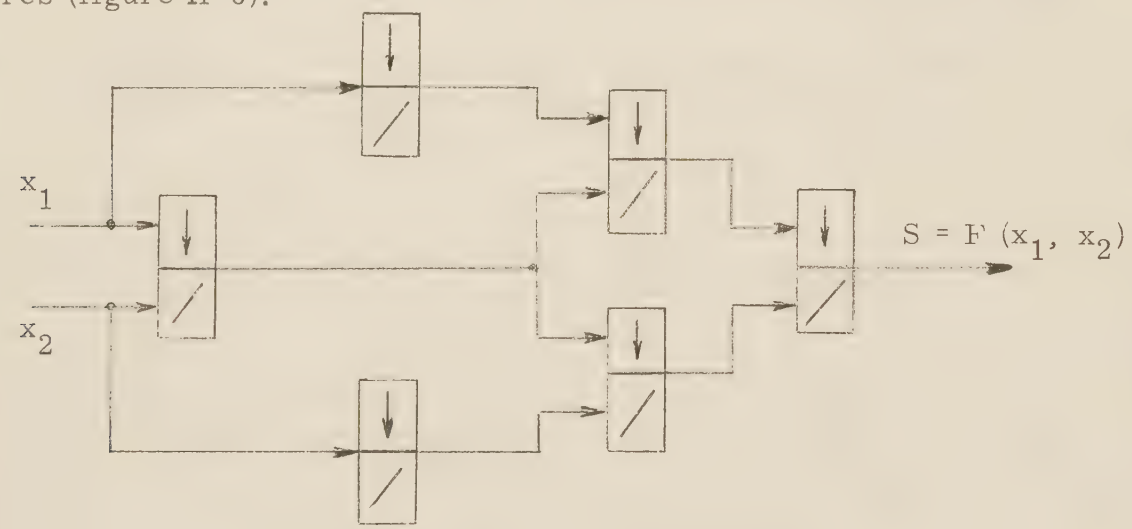


Figure II-6 - Cellule de base modifiée

2. Nous avons vu que la cellule doit réaliser les seize fonctions de ces deux entrées. En fait on sait que trois fonctions seulement sont nécessaires pour caractériser toutes les décompositions des fonctions en combinaison de fonctions à deux arguments. Ces trois fonctions sont le ET, le OU et le OU EXCLUSIF qui sont réalisées par les trois configurations de la figure II-7.

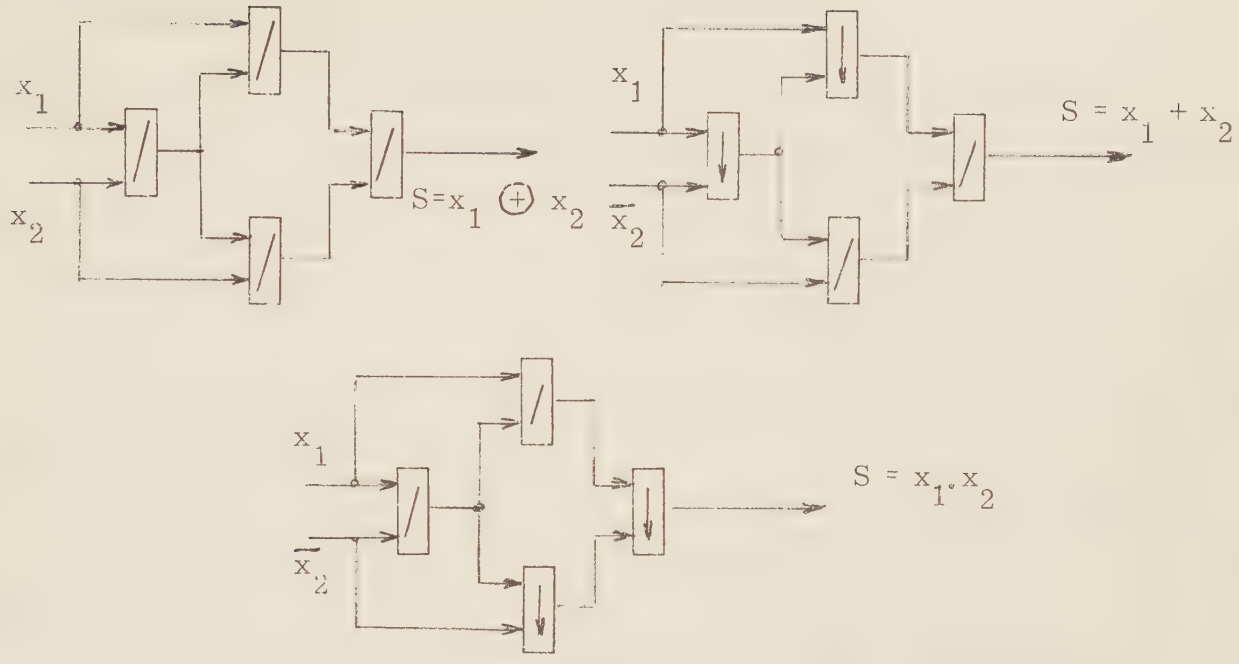


Figure II-7 - Réalisation des fonctions ET, OU et OU EXCLUSIF à l'aide de la cellule de base

II-1-3 - Méthodes d'énumération des fonctions réalisables par une structure en cascade [21]

Le problème qui se pose maintenant est d'utiliser la cellule de base définie dans le paragraphe précédent dans l'élaboration de structures cellulaires en cascade.

Chaque cellule de base pouvant produire toutes les fonctions de deux variables il est intéressant de savoir le nombre de fonctions réalisées par l'association de plusieurs cellules en cascade. Nous décrirons ci-dessous deux méthodes d'énumération l'une directe, l'autre basée sur un théorème énoncé et démontré par Shen.

II-1-3-1 - Première méthode d'énumération

Etudions d'abord le nombre de fonctions de trois variables réalisables par une cascade de deux cellules de base (figure II-8).

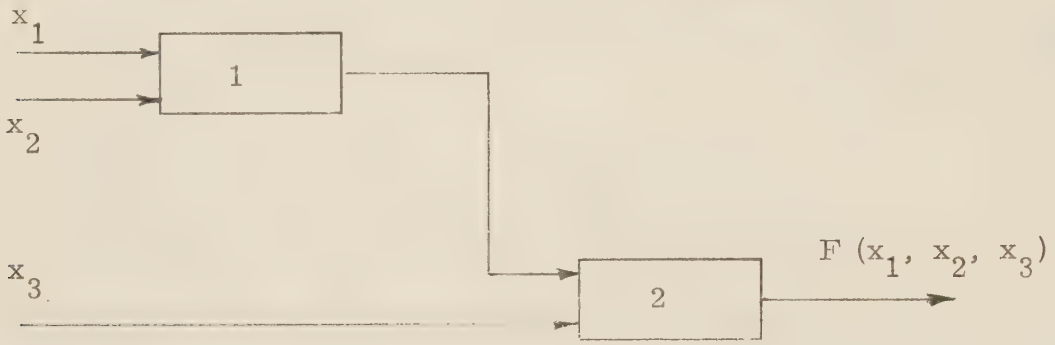


Figure II-8 - Cascade de deux cellules

Ce problème n'est autre qu'un problème de décomposition fonctionnelle. En effet, il s'agit de trouver le nombre de fonctions de trois variables qui peuvent s'écrire sous la forme d'une décomposition disjointe simple :

$$F(x_1, x_2, x_3) = h\left[\varphi(x_1, x_2),\ x_3\right]$$

h étant la fonction réalisée par le module 2

φ étant la fonction réalisée par le module 1

Considérons d'abord une seule cellule : elle réalise toutes les fonctions $\varphi(x_1, x_2)$ de ses deux variables d'entrée x_1 et x_2. Ajoutons une deuxième cellule : sa sortie sera une fonction de x_3 et de $\varphi(x_1, x_2)$. Il existe seize fonctions $\varphi(x_1, x_2)$; x_3 peut prendre les valeurs 0 et 1. Les fonctions de trois variables réalisables par l'ensemble des deux cellules peuvent se grouper en six classes différentes.

Classe 1 : Fonctions $F(x_1, x_2, x_3)$ telles que $F(x_1, x_2, 0) = 0$ quels que soient x_1 et x_2. Cette classe sera notée

$$A_0 = \left\{ x_1 x_2 \ ; \ F(x_1, x_2, 0) = 0 \right\}$$

Elle comporte quatorze fonctions parmi lesquelles $\bar{x}_1 x_3$, $(x_1 \oplus x_2) x_3$

Classe 2 : Fonctions $F(x_1, x_2, x_3)$ telles que $F(x_1, x_2, 0) = 1$ quels que soient x_1, x_2. Cette classe sera notée

$$A_1 = \left\{ x_1, x_2 \ ; \ F(x_1, x_2, 0) = 1 \right\}$$

Elle comporte quatorze fonctions parmi lesquelles : $(\bar{x}_1 \oplus x_2) + \bar{x}_3$

Classe 3 : Fonctions $F(x_1, x_2, x_3)$ telles que $F(x_1, x_2, 1) = 0$ quels que soient x_1, x_2. Cette classe sera notée

$$B_0 = \left\{ x_1, x_2 \ ; \ F(x_1, x_2, 1) = 0 \right\}$$

Elle comporte quatorze fonctions parmi lesquelles $x_1 \cdot x_2 \cdot \bar{x}_3$, $(x_1 \oplus x_2) \bar{x}_3$

Classe 4 : Fonctions $F(x_1, x_2, x_3)$ telles que $F(x_1, x_2, 1) = 1$ quels que soient x_1, x_2. Cette classe sera notée

$$B_1 = \left\{ x_1, x_2 \ ; \ F(x_1, x_2, 1) = 1 \right\}$$

Elle comporte quatorze fonctions parmi lesquelles : $x_1 + \bar{x}_2 + x_3$, $(\bar{x}_1 \oplus x_2) + x_3$

Classe 5 : Fonctions indépendantes de x_3 : ce sont les seize fonctions de x_1 et x_2. Cette classe sera notée :

$$A_0 = B_0$$

$$A_1 = B_1$$

En effet ces fonctions sont telles que les couples x_1, x_2 pour lesquels $F(x_1, x_2, 0) = 0$ sont les mêmes que les couples x_1, x_2 pour lesquels $F(x_1, x_2, 1) = 0$. De plus les couples x_1, x_2 pour lesquels $F(x_1, x_2, 0) = 1$ sont les mêmes que les couples x_1, x_2 pour lesquels $F(x_1, x_2, 1) = 1$.

Classe 6 : Cette classe sera notée :

$$A_0 = B_1$$

$$A_1 = B_0$$

En effet ces fonctions sont telles que les couples $x_1\ x_2$ pour lesquels $F(x_1, x_2, 0) = 0$ sont les mêmes que les couples $x_1\ x_2$ pour lesquels $F(x_1, x_2, 1) = 1$ et les couples $x_1\ x_2$ pour lesquels $F(x_1, x_2, 0) = 1$ sont les mêmes que les couples $x_1\ x_2$ pour lesquels $F(x_1, x_2, 1) = 0$.

Exemple : La fonction $x_1\ x_2 \oplus x_3$ appartient à cette classe. On a :

$$F(x_1, x_2, 0) = x_1\ x_2$$

$$F(x_1, x_2, 1) = \overline{x}_1 + \overline{x}_2$$

$F(x_1, x_2, 0) = 0$ pour les couples $x_1\ x_2$ $\begin{cases} (0\ 0) \\ (0\ 1) \\ (1\ 0) \end{cases}$

$F(x_1, x_2, 0) = 1$ pour les couples $x_1\ x_2$ $(1\ 1)$

$F(x_1, x_2, 1) = 0$ pour les couples $x_1\ x_2$ $(1\ 1)$

$F(x_1, x_2, 1) = 1$ pour les couples $x_1\ x_2$ $\begin{cases} (0\ 0) \\ (0\ 1) \\ (1\ 0) \end{cases}$

Remarque :

En fait les classes 1, 2, 3 et 4 comportent chacune seize fonctions. Parmi ces seize fonctions deux sont déjà comptées dans les classes 5 et 6 ce qui explique le nombre quatorze.

En définitive le nombre N_3 de fonctions de trois variables réalisables par une cascade de deux cellules est :

(1) $N_3 = 2 \times 16 + 4\,(16 - 2) = 88$

Remarques :

1. Parmi ces 88 fonctions, 50 seulement sont non dégénérées

2. Seules des cascades irrédondantes sont étudiées. De plus les différentes entrées libres des cascades ne sont pas permutables ce qui réduit le nombre de fonctions réalisables.

Généralisation de la formule 1 : Cas d'une cascade de n - 1 cellules (n entrées)

Supposons que la formule (1) s'applique au cas d'une cascade de n - 2 cellules et de n - 1 entrées. Le nombre de fonctions réalisables par une telle cascade est alors

$$(2) \quad N_{n-1} = 2\,N_{n-2} + 4\,(N_{n-2} - 2)$$

Montrons que N_n vérifie une relations du type (2) (figure II-9).

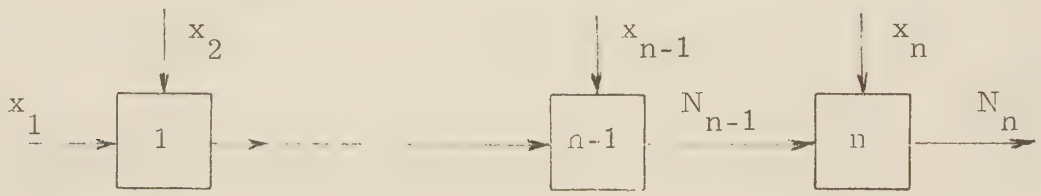


Figure II-9

La cellule n reçoit comme entrées d'une part N_{n-1} fonctions (N_{n-1} donnée par (2)), d'autre part la variable x_n. Le problème est alors identique au cas de la cascade de deux cellules traitée précédemment. Les N_n fonctions vont se répartir en six classes. Les classes 1, 2, 3, 4 contiendront chacune (N_{n-1} -2) fonctions, les classes 5 et 6, N_{n-1} fonctions. Le nombre de fonctions réalisables par une cascade de n-1 cellules sera donc :

$$N_n = 2\,N_{n-1} + 4\,(N_{n-1} - 2)$$

c'est-à-dire

$$(3) \quad N_n = 6\,N_{n-1} - 8$$

La relation (3) permettra donc de calculer le nombre de fonctionsréalisables par une cascade de n-1 cellules si l'on prend la condition initiale $N_1 = 4$.

Remarques

1. La formule (3) n'est valable que dans le cas d'une cascade irrédondante.

2. Nous avons donné ici une méthode pour énumérer les classes de fonctions réalisables. Le résultat contenu dans la relation (3) peut aussi être retrouvé par d'autres approches et en particulier en raisonnant sur des notions d'ensembles et d'applications entre ensembles.

II-1-3-2 - Deuxième méthode d'énumération

Cette méthode est basée sur la notion de dérivées booléennes d'une fonction logique dont nous donnons la définition et quelques propriétés ci-dessous.

Définition : La dérivée booléenne d'une fonction de n variables $F(x_1, x_2, \ldots, x_n)$ par rapport à la variable x_i est définie par :

$$\frac{\partial F}{\partial x_i} = F'_{x_i} = F(x_1, x_2, \ldots, 0, \ldots, x_n) \oplus F(x_1, x_2, \ldots, 1, \ldots, x_n) = F_i$$

Propriétés de la dérivée booléenne

1 $$\frac{\partial}{x_i}[f + g] = [f + g]_i = f_i + g_i$$

2 $$\frac{\partial}{x_1}[x_1 . f(x_2, x_3, \ldots, x_n)] = f(x_2, x_3, \ldots, x_n) = \frac{\partial}{x_1}[x'_1 f(x_2, x_3, \ldots, x_n)]$$

3 $$f_{ij}(x_1, x_2, \ldots, x_n) = f_{ji}(x_1, x_2, \ldots; x_n)$$

4. Théorème (Différentiation des fonctions implicites)

Soit $\{A, B\}$ une partition de l'ensemble X des variables. Si $f(X) = F(\varphi(A), B)$ alors

$$\frac{\partial f}{\partial x} = \frac{\partial F}{\partial \varphi} \cdot \frac{\partial \varphi}{\partial x} \qquad \text{pour tout } x \in A$$

5. Jacobien [22] de deux fonctions f, g de n variables par rapport à un sous ensemble (x_i, x_j) de X

$$J\left(\frac{f, g}{x_i, x_j}\right) = \begin{vmatrix} f_i & f_j \\ g_i & g_j \end{vmatrix} = f_i g_j \oplus f_j g_i$$

Nous avons vu que la recherche des fonctions réalisables par une cascade de cellules se ramenait en fait à la recherche des décompositions disjointes simples de cette fonction. Le théorème suivant dû à Shen donne un moyen simple et élégant de trouver ces décompositions dans un cas particulier.

Théorème : Si

$$f(x_1, x_2, x_3) = F[\phi(x_1, x_2), x_3] \qquad \text{alors} \qquad J\left(\frac{f, \phi}{x_1, x_2}\right) = 0$$

quels que soient x_1 et x_2

En effet :

$$J\left(\frac{f,\ \phi}{x_1, x_2}\right) = \begin{vmatrix} f_1 & f_2 \\ \phi_1 & \phi_2 \end{vmatrix} = \begin{vmatrix} F_\phi \cdot \phi_1 & F_\phi \cdot \phi_2 \\ \phi_1 & \phi_2 \end{vmatrix} = 0$$

Ce théorème donne donc une condition nécessaire de décomposition.

Corollaire : Condition suffisante de décomposition

S'il existe une fonction $\phi\ (x_1,\ x_2)$ telle que $J\left(\frac{f,\ \phi}{x_1, x_2}\right) = 0$ alors il existe une fonction $F\ (y,\ x_3)$ telle que $f\ (x_1, x_2, x_3) = F\left[\phi\ (x_1,\ x_2),\ x_3\right]$

Ce corollaire va nous permettre de vérifier les résultats obtenus sur le nombre de fonctions réalisables.

Toute fonction $f\ (x_1,\ x_2,\ x_3)$ de trois variables peut s'écrire :

$$f(x_1, x_2, x_3) = b_0 \oplus b_1\ x_1 \oplus b_2\ x_2 \oplus b_3\ x_3 \oplus b_{12}\ x_1 x_2 \oplus b_{13}\ x_1 x_3 \oplus b_{23}\ x_2 x_3 \oplus b_{123}\ x_1 x_2 x_3$$

Cette expression peut se mettre sous la forme :

$$f\ (x_1, x_2, x_3) = H \oplus x_1\ G^1 \oplus x_2\ G^2 \oplus x_1\ x_2\ G^3$$

avec

$$H = b_0 \oplus b_3\ x_3$$
$$G^1 = b_1 \oplus b_{13}\ x_3$$
$$G^2 = b_2 \oplus b_{23}\ x_3$$
$$G^3 = b_{12} \oplus b_{123}\ x_3$$

La condition nécessaire et suffisante pour que F existe est que les G^i non nuls soient égaux.

On a : $\phi(x_1, x_2) = a_0 \oplus a_1 x_1 \oplus a_2 x_2 \oplus a_{12} x_1 x_2$

La condition $J\left(\frac{f, \phi}{x_1, x_2}\right) = 0$ implique

$$\begin{vmatrix} f_1 & f_2 \\ \phi_1 & \phi_2 \end{vmatrix} = \begin{vmatrix} G^1 \oplus G^3 x_2 & G^2 \oplus G^3 x_1 \\ a_1 \oplus a_{12} x_2 & a_2 \oplus a_{12} x_1 \end{vmatrix} = 0$$

c'est-à-dire $(G^1 \oplus G^3 x_2)(a_2 \oplus a_{12} x_1) + (G^2 \oplus G^3 x_1)(a_1 \oplus a_{12} x_2) = 0$

Cette équation doit être vérifiée quels que soient x_1 x_2, c'est-à-dire

x_1	x_2		
0	0	$G^1 a_2 + G^2 a_1 = 0$	(a)
0	1	$G^3 a_2 + G^2 a_{12} = 0$	(b)
1	0	$G^1 a_{12} + G^3 a_1 = 0$	(c)
1	1	vérifiée si (a), (b) et (c) sont vraies.	

En transposant notre raisonnement sur les coefficients a_i on obtient les conditions que doivent vérifier les fonctions réalisables.

a_1	a_2	a_3	$\phi(x_1, x_2)$		Fonctions réalisables
0	0	0	a_0	$G^1 = G^2 = G^3$	$b_0 \oplus b_3 x_3$
0	0	1	$a_0 \oplus x_1 x_2$	$G^1 = G^2 = 0$	$b_0 \oplus b_3 x_3 \oplus x_1 x_2 (b_{12} \oplus b_{123} x_3)$
0	1	0	$a_0 \oplus x_2$	$G^1 = G^3 = 0$	$b_0 \oplus b_3 x_3 + (b_2 \oplus b_{23} x_3) x_2$
0	1	1	$a_0 \oplus x_2 \oplus x_1 x_2$	$G^1 = 0$ $G^2 = G^3$	$b_0 \oplus b_3 x_3 \oplus (b_2 \oplus b_{23} x_3) (x_2 \oplus x_1 x_2)$
1	0	0	$a_0 \oplus x_1$	$G^2 = G^3 = 0$	$b_0 \oplus b_3 x_3 \oplus (b_1 \oplus b_{13} x_3) x_1$
1	0	1	$a_0 \oplus x_1 \oplus x_1 x_2$	$G^2 = 0$ $G^1 = G^3$	$b_0 \oplus b_3 x_3 \oplus (b_1 \oplus b_{13} x_3) (x_1 \oplus x_1 x_2)$
1	1	0	$a_0 \oplus x_1 \oplus x_2$	$G^3 = 0$ $G^1 = G^2$	$b_0 \oplus b_3 x_3 \oplus (b_1 \oplus b_{13} x_3) (x_1 \oplus x_2)$
1	1	1	$a_0 \oplus x_1 \oplus x_2 \oplus x_1 x_2$	$G^1 = G^2 = G^3$	$b_0 \oplus b_3 x_3 \oplus (b_1 \oplus b_{13} x_3) (x_1 \oplus x_2 \oplus x_3)$

Table I

On retrouve effectivement ainsi les fonctions réalisables par une cascade de deux cellules. Il est clair que la généralisation de cette méthode d'énumération suppose des études plus poussées sur les propriétés du Jacobien qui sortent du cadre de nos préoccupations.

II-1-4 - Tests permettant de déterminer si une fonction est réalisable par une structure en cascade

Nous avons déterminé dans le paragraphe précédent le nombre de fonctions réalisables par une structure en cascade. Le problème est alors de savoir quand est-ce qu'une fonction booléenne est réalisable par une telle structure.

Pour répondre à cette question nous avons mis au point une série de tests s'appuyant sur la définition des différentes classes. Ces tests font appel à la notion de dérivée booléenne que nous avons définie précédemment.

II-1-4-1 - Définition des tests

Reprenant l'énumération des six classes du paragraphe précédent, et raisonnant pour le cas de trois variables les tests suivants peuvent être définis.

Classe 1 : Si $F(x_1, x_2, 0) = 0 \ \forall \ x_1 \ x_2$ alors $F(x_1, x_2, x_3) = F'_{x_3} \ x_3$

Classe 2 : Si $F(x_1, x_2, 0) = 1 \ \forall \ x_1 \ x_2$ alors $F(x_1, x_2, x_3) = \overline{F'_{x_3}} + \overline{x_3}$

Classe 3 : Si $F(x_1, x_2, 1) = 0 \ \forall \ x_1 \ x_2$ alors $F(x_1, x_2, x_3) = F'_{x_3} \cdot \overline{x_3}$

Classe 4 : Si $F(x_1, x_2, 1) = 1 \ \forall \ x_1 \ x_2$ alors $F(x_1, x_2, x_3) = \overline{F'_{x_3}} + x_3$

Classe 5 Une fonction appartient à cette classe si $F'_{x_3} = 1$. Elle est de la forme

$$F(x_1, x_2, x_3) = F(x_1, x_2, 0)$$

Classe 6 Une fonction appartient à cette classe si $F'_{x_3} = 1$. Elle est de la forme

$$F(x_1, x_2, x_3) = x_3 \oplus F(x_1, x_2, 0)$$

Cette série de tests permet de déterminer si la fonction est réalisable par une cascade de façon relativement simple. Remarquons néanmoins que lorsque le nombre de variables augmente, les tests peuvent devenir compliqués et il est alors préférable d'utiliser la programmation.

II-1-4-2 - Exemple d'application

Soit à réaliser à l'aide d'une structure en cascade la fonction de 4 variables

$$F(x_1, x_2, x_3, x_4) = x_3\bar{x}_4 + x_1\bar{x}_2\bar{x}_4 + \bar{x}_1x_2\bar{x}_4 + \bar{x}_1\bar{x}_2\bar{x}_3x_4 + x_1x_2\bar{x}_3x_4$$

Appliquant la série de tests on trouve : $F'_{x_4} = 1$

Donc :

$$F(x_1, x_2, x_3, x_4) = x_4 \oplus F(x_1, x_2, x_3, 0)$$

$$F(x_1, x_2, x_3, 0) = \bar{x}_3 + x_1\bar{x}_2 + \bar{x}_1x_2$$

On a : $F(x_1, x_2, 0, 0) = 1$ qui correspond bien à la décomposition trouvée.

La fonction $x_1 \bar{x}_2 + \bar{x}_1 x_2 = x_1 \oplus x_2$ est bien telle que $F'_{x_2} = 1$.

La cascade correspondant à $F(x_1, x_2, x_3, x_4)$ est représentée figure II-10.

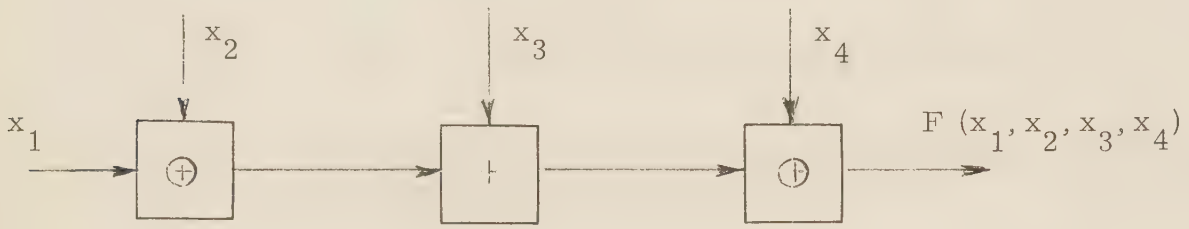


Figure II-10 - Cascade réalisant une fonction de quatre variables

II-2 - SYNTHESE DE FONCTIONS LOGIQUES A L'AIDE D'UN RESEAU CELLULAIRE BIDIMENSIONNEL

Nous avons abordé dans le premier chapitre les problèmes qui se posent au niveau de l'analyse des réseaux cellulaires et de la synthèse des fonctions logiques à l'aide de ce type de structures. D'une façon générale nous pensons que les difficultés majeures posées par ces réseaux se placent à trois niveaux :

- choix d'une cellule de base de complexité raisonnable : il serait en effet inopportun de vouloir arriver à une structure d'interconnexion itérative et régulière sans tenir aucun compte de la complexité de la cellule de base elle-même.

- élaboration de méthodes de synthèse systématiques qui permettront de généraliser l'emploi de ces réseaux.

- minimisation des dimensions du réseau et du nombre de connexions externes qui sont deux paramètres importants au niveau d'une réalisation technologique.

Ce paragraphe, consacré à l'étude d'un réseau cellulaire particulier, apporte une contribution à la résolution de ces différents problèmes.

II-2-1 - Description de la structure choisie

Nous avons choisi la structure de réseau cellulaire représentée sur la figure II-11.

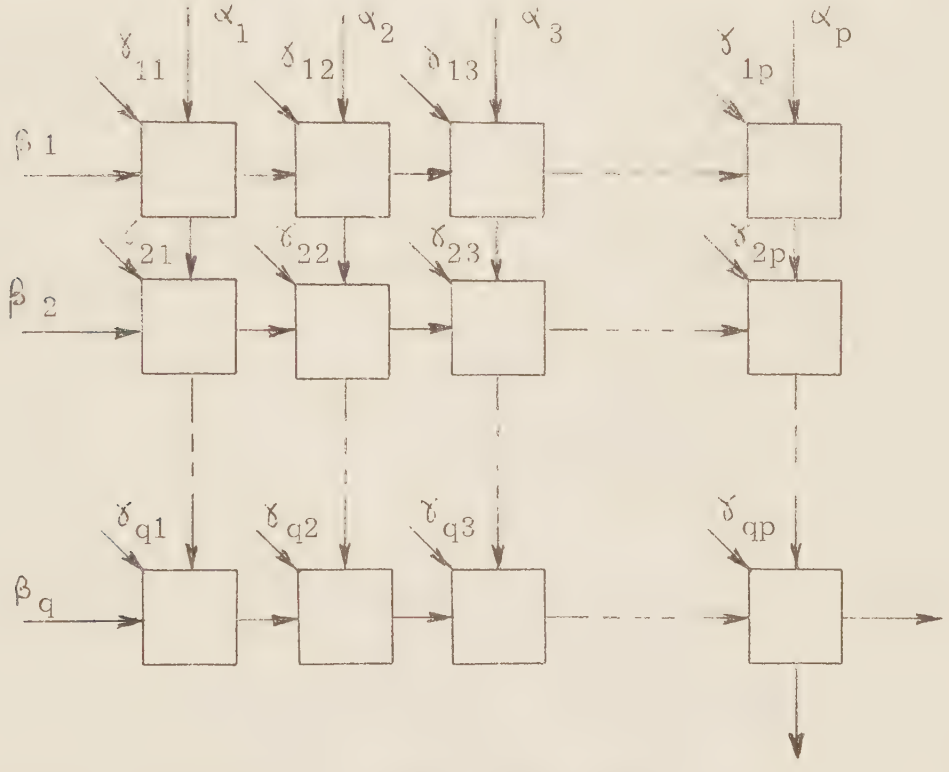


Figure II-11 - Structure du réseau

Notre choix est motivé avant tout par la simplicité de la structure d'interconnexion de ce réseau rectangulaire.

Les dimensions du réseau sont définies par :

- le nombre de cellules N = p x q, p représentant la largeur du réseau et q sa hauteur.

-le nombre de connexions externes $\alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q$ et $\gamma_{11}, \ldots, \gamma_{pq}$: C = p + q + pq

Le problème que nous nous proposons de résoudre est celui de la synthèse d'une fonction logique de n variables $F(x_1, x_2, \ldots, x_n)$ à l'aide d'un réseau cellulaire de ce type et de dimensions minimales.

La fonction $F(x_1, x_2, \ldots, x_n)$ sera réalisée en sortie de la cellule correspondant aux valeurs maximales de p et q pour le réseau étudié.

Les entrées externes $\alpha_1, \alpha_2, \ldots, \alpha_p, \beta_1, \beta_2, \ldots, \beta_q$ et $\gamma_{11}, \gamma_{12}, \ldots, \gamma_{pq}$ seront choisies dans l'ensemble

$$\{0, 1, x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$$

II-2-2 - Description de la cellule de base du réseau

La cellule de base est schématisée sur la figure II-12.

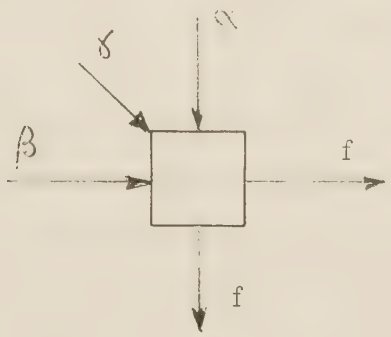


Figure II-12 - Cellule de base

C'est un opérateur à trois entrées α, β, γ et deux sorties qui réalisent la même fonction des entrées α, β, γ : f. Ce dédoublement de la sortie est imposé par la structure étudiée.

La fonction f a pour expression logique :

$$f = \gamma \bar{\beta} + \gamma' \alpha \qquad (4)$$

II-2-3 - Propriétés de la cellule de base

a - C'est un opérateur complet et totalement asymétrique.

b - Il réalise une fonction de trois variables de complexité raisonnable donc il répond au premier critère défini dans l'introduction de ce paragraphe. Nous donnons ci-dessous deux réalisations minimales de cet opérateur à l'aide de ON et de NI.

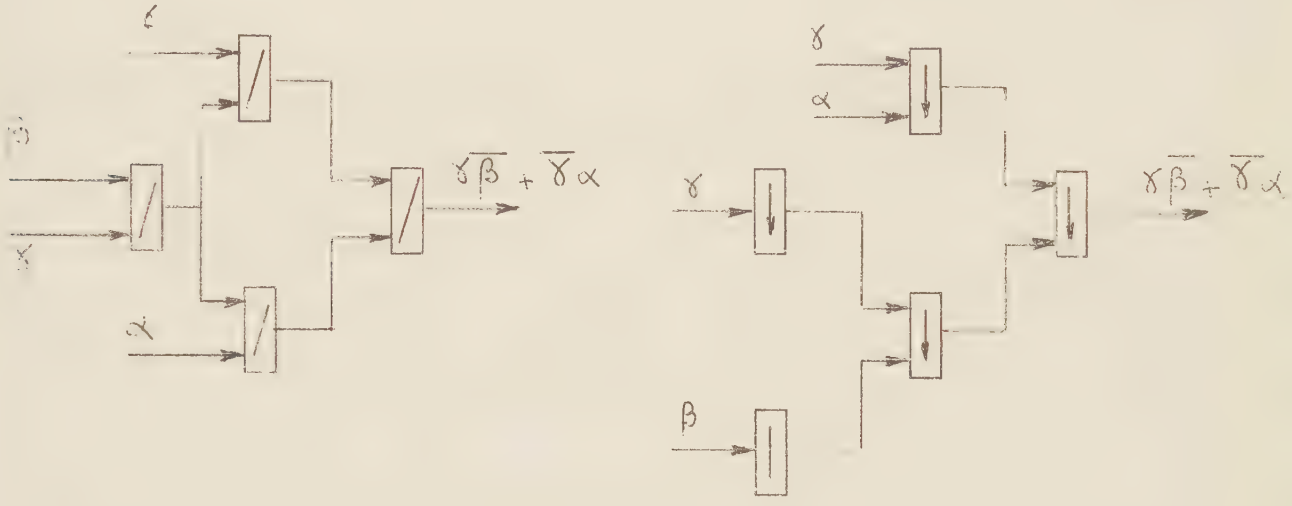


Figure II-13 - Deux réalisations de la cellule de base en opérateurs NI et ON

c - Le complément de f est défini par : $\bar{f} = \gamma\beta + \bar{\gamma}\bar{\alpha}$ et est obtenu à partir de f en complémentant les entrées α et β .

d - Si $\gamma = 0$ $\quad f = \alpha$

Si $\gamma = 1$ $\quad f = \bar{\beta}$

Donc en assignant l'entrée γ à une valeur constante particulière, on peut soit propager une information de haut en bas, soit propager le complément d'une information de la gauche vers la droite. Nous utiliserons cette propriétés intéressante pour faire la synthèse d'une fonction logique de n variables (dans le cas $n \geq 3$)

Ces définitions et remarques préliminaires sont suffisantes pour aborder le problème essentiel qui nous préoccupe, celui de la synthèse d'une fonction logique de n variables.

Notre approche ira du particulier en général, c'est en ce sens que nous commencerons par étudier un réseau de dimensions réduites, hauteur 2, largeur 2, nombre d'entrées externes : 8. Nous nous servirons ensuite des résultats acquis au niveau de ce cas restrictif pour énoncer une méthode de synthèse des fonctions de n variables.

II-2-4 - Etude d'un réseau de dimensions N = 4, C = 8. Synthèse d'une fonction de trois variables

Nous montrerons d'abord que l'on peut réaliser une fonction arbitraire de trois variables sur n'importe quelle partie du réseau général de dimension N = 4, C = 8 à condition que les entrées soient indépendantes et que chacune d'elles puisse être choisie dans l'ensemble $\{0, 1, x, \bar{x}\}$

Lemme : La cellule de base du réseau décrite dans les paragraphes II-2-2 et II-2-3 réalise toutes les fonctions booléennes de deux variables.

En effet, la relation (4) $f = \gamma\bar{\beta} + \bar{\gamma}\alpha$ représente la décomposition de Shannon d'une fonction de trois variables α, β, γ par rapport à la variable γ. Si les entrées α et β de la cellule sont indépendantes et prennent leurs valeurs uniquement dans l'ensemble $\{0, 1, \delta, \bar{\delta}\}$ (δ variable booléenne) la relation (4) peut représenter suivant le choix de α et β dans $\{0, 1, \delta, \bar{\delta}\}$ l'expression logique de toute fonction booléenne des deux variables γ et δ.

Considérons alors la structure représentée sur la figure II-14 qui est un sous-réseau de dimensions N = 4, C = 8 du réseau général de la figure II-11.

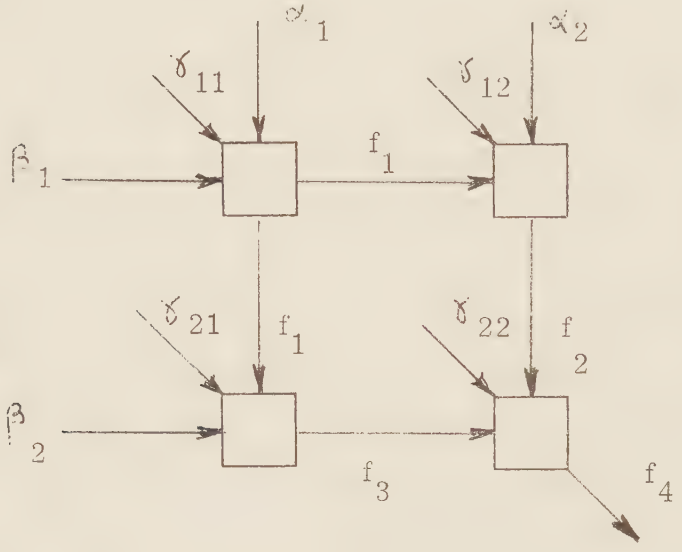


Figure II-14 - Réseau à quatre cellules

Son intérêt est caractérisé par le théorème suivant :

Théorème 1 : Le réseau de figure II-14 réalise, pour un choix convenable de ses entrées externes $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_{11}, \gamma_{12}, \gamma_{21}, \gamma_{22}$ dans l'ensemble $\{0, 1, x_1, \bar{x}_1, x_2, \bar{x}_2, x_3, \bar{x}_3\}$ toute fonction logique de trois variables x_1, x_2, x_3.

En effet, cherchons l'expression de f_4 en fonction des entrées externes, nous avons :

(5) $f_4 = \gamma_{22}\,\bar{f}_3 + \bar{\gamma}_{22}\,f_2$

(6) $f_3 = \gamma_{21}\,\bar{\beta}_2 + \bar{\gamma}_{21}\,f_1$

(7) $f_2 = \gamma_{12}\,\bar{f}_1 + \overline{\gamma_{12}}\,\alpha_2$

(8) $f_1 = \gamma_{11}\,\overline{\beta_1} + \bar{\gamma}_{11}\,\alpha_1$

d'où :

(9) $f_4 = \gamma_{22}\left[\bar{\gamma}_{21}\left(\bar{\gamma}_{11}\,\bar{\alpha}_1 + \gamma_{11}\,\beta_1\right) + \gamma_{21}\,\beta_2\right] +$

$+\bar{\gamma}_{22}\left[\gamma_{12}\left(\bar{\gamma}_{11}\,\bar{\alpha}_1 + \gamma_{11}\,\beta_1\right) + \overline{\gamma_{12}}\,\beta_2\right]$

Si nous prenons $\gamma_{11} = \gamma_{12} = \gamma_{21} = x_2$ et $\gamma_{22} = x_1$ l'expression (9) devient :

(10) $f_4 = x_1\left[\bar{x}_2\,\bar{\alpha}_1 + x_2\,\beta_2\right] + \bar{x}_1\left[x_2\,\beta_1 + \bar{x}_2\,\alpha_2\right]$

La relation (10) représente le développement de Shannon de la fonction f_4 par rapport aux variables x_1 et x_2.

Si les entrées α_1, α_2 et β_1, β_2 sont indépendantes et sont choisies dans l'ensemble $\{0,\ 1,\ x_3,\ \bar{x}_3\}$ la relation (10) n'est autre que la décomposition de Shannon d'une fonction de trois variables.

En effet, (10) s'écrit :

(11) $f_4 = x_1\left[\bar{x}_2\,y_1\,(0,\ 1,\ x_3,\ \bar{x}_3) + x_2\,y_2\,(0,\ 1,\ x_3,\ \bar{x}_3)\right]$

$+\bar{x}_1\left[x_2\,y_3\,(0,\ 1,\ x_3,\ \bar{x}_3) + \bar{x}_2\,y_4\,(0,\ 1,\ x_3,\ \bar{x}_3)\right]$

Si nous choisissons de façon adéquate y_1, y_2, y_3 et y_4 le réseau de la figure II-14 pourra réaliser simplement toute fonction de trois variables.

Exemples :

a - Fonction parité de trois variables : $F = x_1 \oplus x_2 \oplus x_3$

Cette fonction peut s'écrire $F = x_1 (\bar{x}_2 \bar{x}_3 + x_2 x_3) + \bar{x}_1 (x_2 \bar{x}_3 + \bar{x}_2 x_3)$

Le réseau qui réalise cette fonction est donné figure II-15.

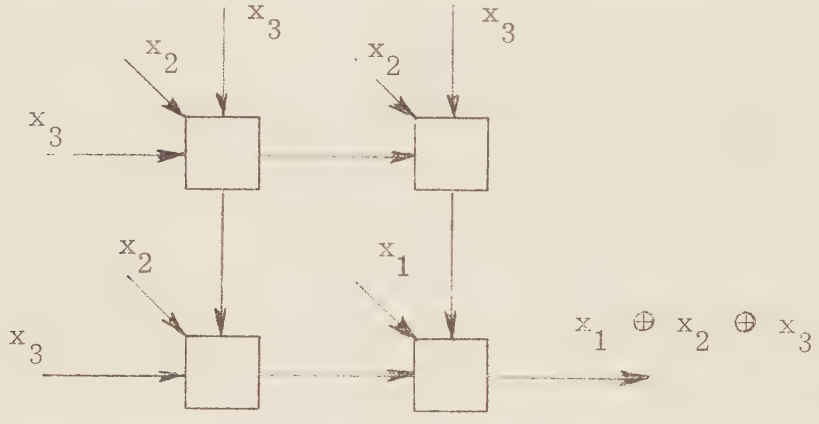


Figure II-15

b - Fonction majorité de trois variables : $F = x_1 x_2 + x_1 x_3 + x_2 x_3$

Cette fonction peut s'écrire : $F = x_1 (\bar{x}_2 x_3 + x_2 . 1) + \bar{x}_1 (x_2 x_3 + x_2 . 0)$

Il lui correspond le réseau de la figure II-16.

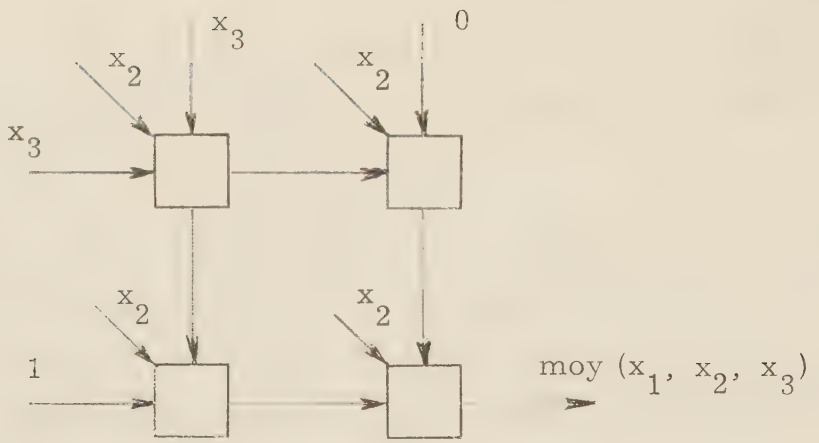


Figure II-16

II-2-5 - Synthèse d'une fonction de n variables

II-2-5-1 - Cas particulier n = 4

La décomposition de Shannon d'une fonction de quatre variables peut se mettre sous la forme :

$$F(x_1, x_2, x_3, x_4) = x_1 F_1(x_2, x_3, x_4) + \bar{x}_1 F_2(x_2, x_3, x_4)$$

Les fonctions F_1 et F_2 sont des fonctions de trois variables qui peuvent être réalisées simplement par une structure du type décrit dans le paragraphe précédent et de dimensions N = 4, C = 8. La réalisation de $F(x_1, x_2, x_3, x_4)$ conduit donc au logigramme de la figure II-17.

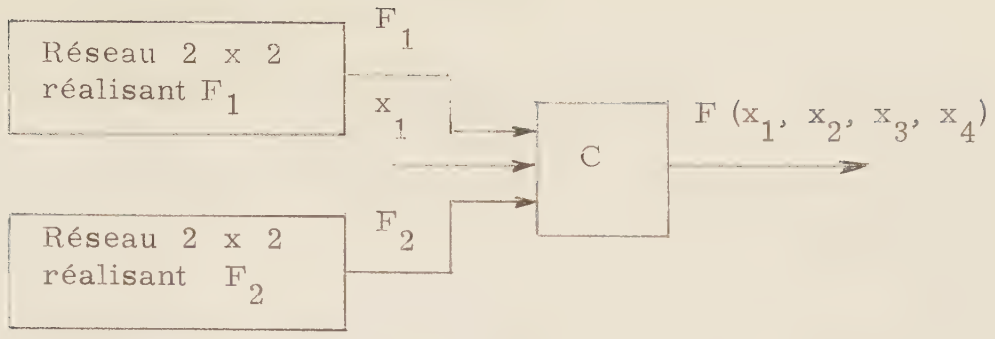


Figure II-17

La cellule C représente une cellule de base identique à celle décrite ci-dessus.

Le problème est alors d'implanter le réseau de la figure II-17 dans un réseau cellulaire ayant la structure d'interconnexion régulière décrite sur la figure II-11. Cette opération est rendue possible par l'utilisation de la propriété du paragraphe II-2-3 qui permet de définir deux cellules de transfert de l'information suivant la valeur de γ.

a - Cas où $\gamma = 0$

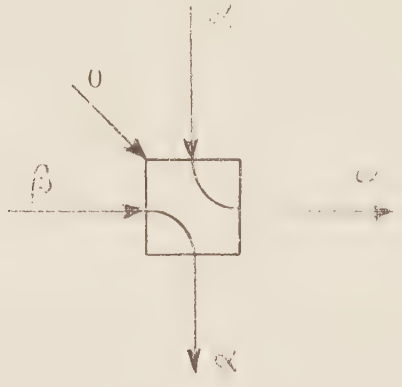


Figure III-18

L'aiguillage de l'information arrivant à l'entrée α se fait quelle que soit la valeur de β .

b - Cas où $\gamma = 1$.

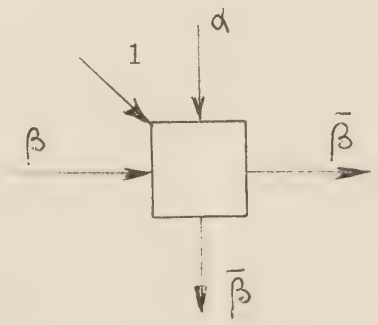


Figure II-19

L'aiguillage du complément de l'information arrivant à l'entrée β se fait quelle que soit la valeur de α .

Ces deux cellules de transfert nous permettent de transformer le réseau de la figure II-17 à structure d'interconnexion inhomogène (cellulaire et arborescent) en un réseau cellulaire homogène (figure II-20), qui réalisera suivant les valeurs de ses entrées externes n'importe quelle fonction de quatre variables.

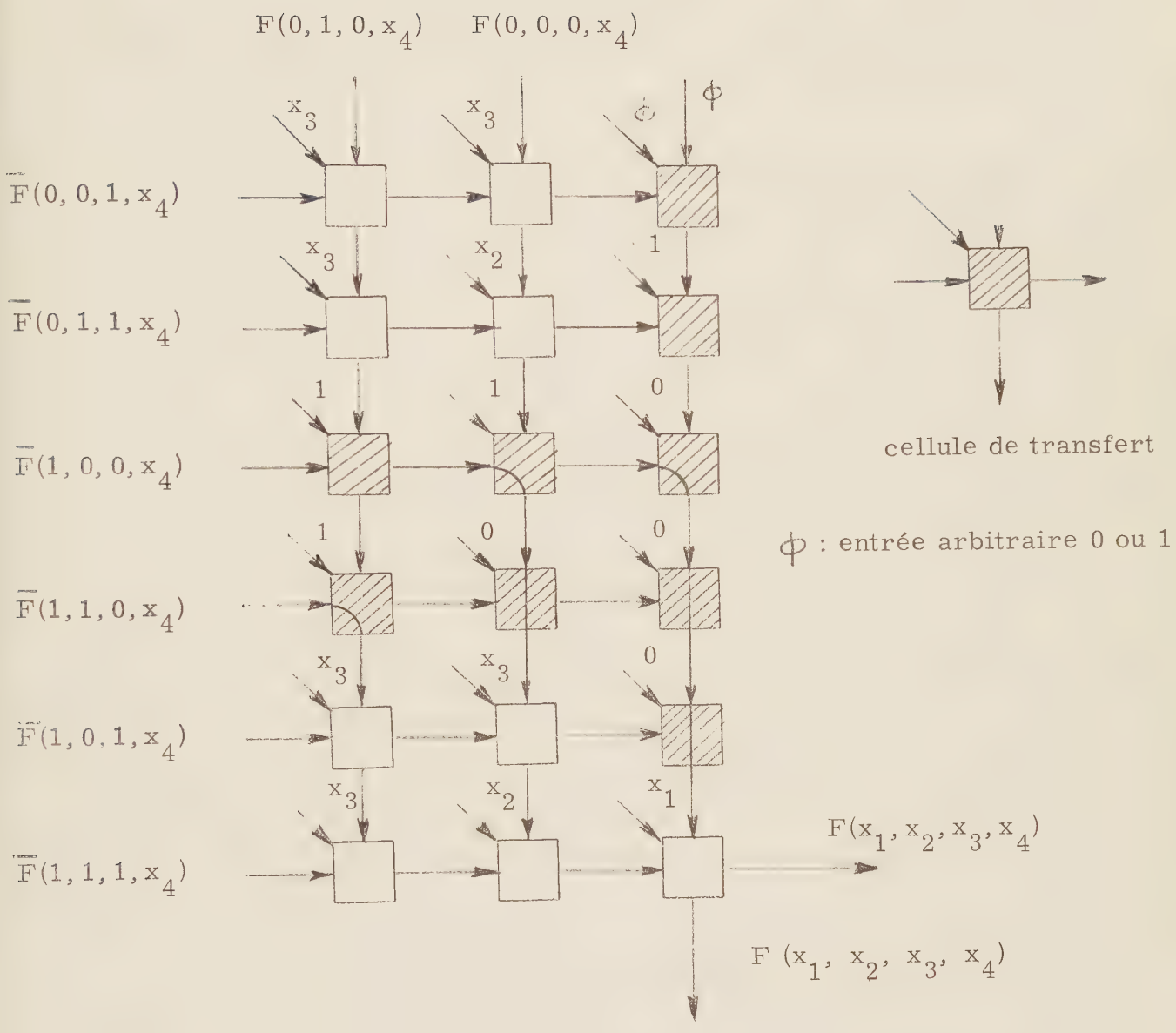


Figure II-20

Les dimensions de ce réseau sont :

hauteur : 6

largeur : 3

nombre de cellules : N = 3 x 6 = 18

Nombre d'entrées externes : C = 5 + 8 = 13

En fait, ce réseau élaboré d'une façon systématique ne répond pas au second critère défini ci-dessus à savoir le critère de minimalité. En effet, attardons nous quelques instants à l'analyse du réseau cellulaire représenté sur la figure II-21.

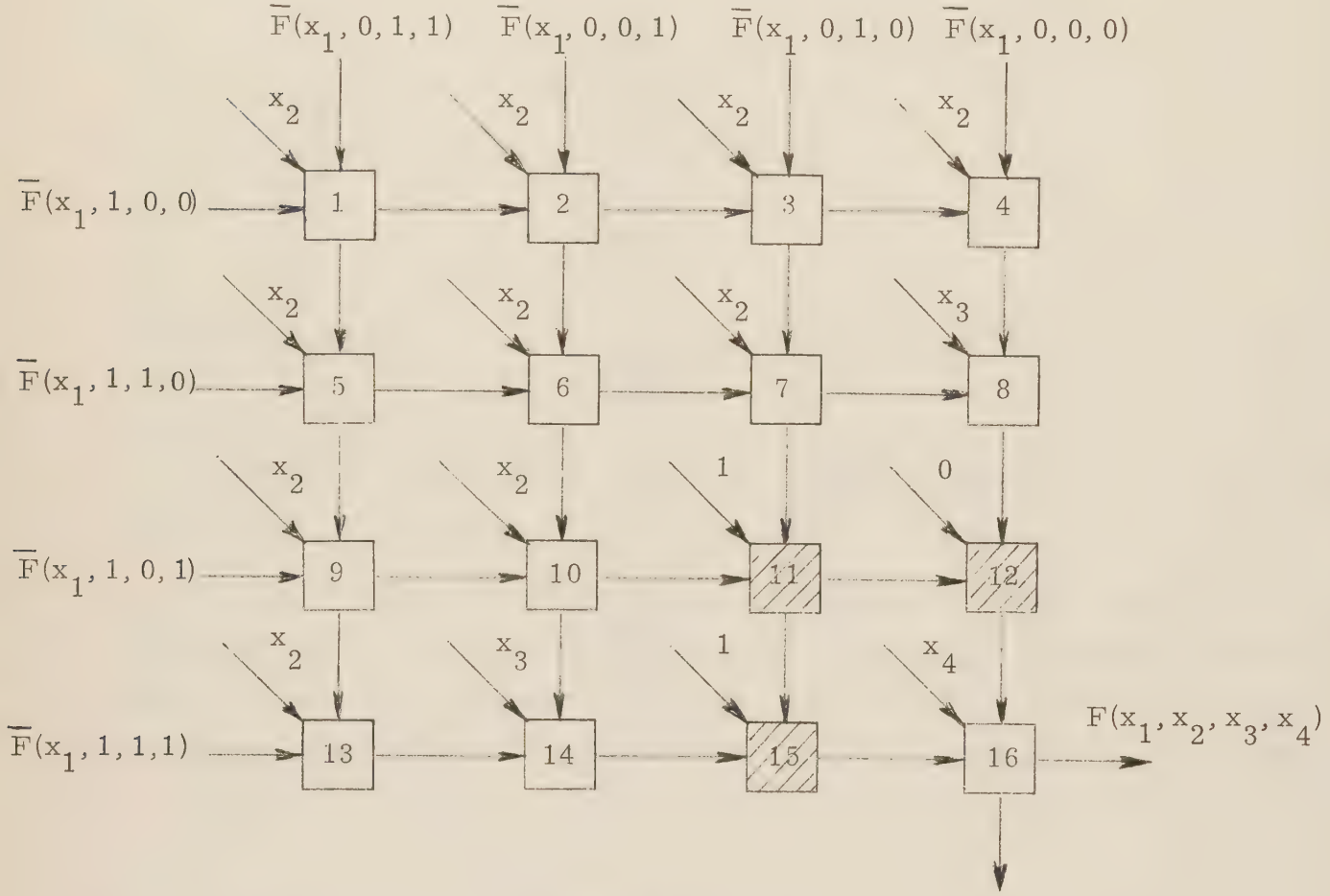


Figure II-21

Nous allons montrer que ce réseau réalise toute fonction de quatre variables. Les fonctions réalisées en sortie de cellules 4, 7, 10 et 13 sont respectivement :

$$f_4 = x_2 \overline{F}(x_1, 1, 0, 0) + \overline{x}_2 F(x_1, 0, 0, 0)$$
$$f_7 = x_2 \overline{F}(x_1, 1, 1, 0) + \overline{x}_2 \overline{F}(x_1, 0, 1, 0)$$
$$f_{10} = x_2 \overline{F}(x_1, 1, 0, 1) + x_2 F(x_1, 0, 1, 1)$$
$$f_{13} = x_2 \overline{F}(x_1, 1, 1, 1) + \overline{x}_2 \overline{F}(x_1, 0, 1, 1)$$

Ces quatre expressions représentent les décompositions de Shannon de quatre fonctions logiques indépendantes des deux variables x_1 et x_2. On peut donc par un assignement adéquat des cellules 8, 11, 12, 15 et 16 réaliser en sortie de la cellule 16 toute fonction logique de quatre variables.

Le réseau de la figure II-21 représente donc le réseau cellulaire minimum composé d'opérateurs $f = \gamma\bar{\beta} + \bar{\gamma}\alpha$ permettant de faire la synthèse de toute fonction de quatre variables.

Ses dimensions sont : hauteur : 4
largeur : 4
nombre de cellules : 16
nombre d'entrées externes : 12

II-2-5-2 - Cas général (n quelconque)

Dans le cas général, la synthèse d'une fonction arbitraire de n variables s'effectue de la manière suivante :

a - on développe sous forme canonique la fonction donnée par rapport à n - 3 de ses variables

b - on assigne les entrées d'un réseau N = 4, C = 8 pour chacune des fonctions de trois variables obtenues par la méthode décrite au paragraphe II-2-4

c - on fait alors la synthèse de 2^{n-4} fonctions de quatre variables en employant la méthode du développement canonique.

d - identique à c pour les 2^{n-5} fonctions de cinq variables, 2^{n-6} fonctions de six variables, on donne figure II-22 la configuration minimale correspondant à une fonction de cinq variables.

La table II donne les indications sur les dimensions et le nombre d'entrées externes des réseaux minimaux pour $n \leq 6$.

Fonctions de 5 variables

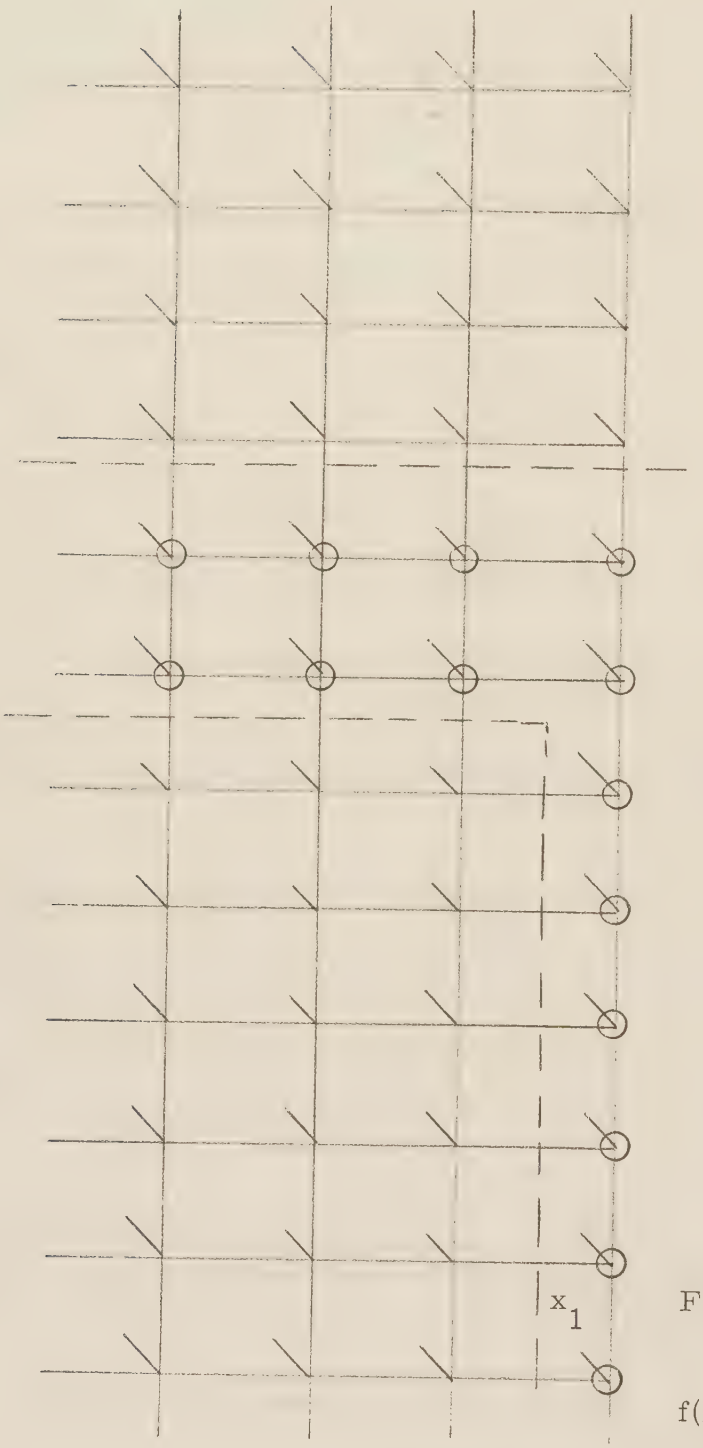


Figure II-22

Nombre de variables	Dimensions du réseau	Nombre d'entrées externes
2	1 x 1	3
3	2 x 2	6
4	4 x 4	13
5	12 x 4	22
6	28 x 5	39

Table II

Remarque

Deux entrées externes ne sont considérées comme distinctes que si elles sont attaquées par deux variables différentes.

On peut montrer que le nombre d'entrées externes pour $n \geqslant 4$ croît comme $2^{n+1} + n + 1$ et que le nombre de cellules de base varie comme $(2^{n-1} - 4) \times (n - 1)$, ce qui représente une amélioration non négligeable par rapport aux résultats déjà obtenus dans ce domaine.

II-3 - CONCLUSIONS DU CHAPITRE II

Nous avons dans ce chapitre présenté deux types de réseaux. Dans le domaine des structures en cascade notre contribution se situe à trois niveaux :

- élaboration d'une cellule de base de structure simple
- description de méthodes d'analyse basées sur un outil booléen simple et non sur des constructions graphiques relativement complexes
- ébauche d'une méthode de synthèse systématique.

En ce qui concerne les réseaux bidimensionnels décrits dans le paragraphe II-2 nous pensons avoir atteint au moins partiellement les objectifs généraux que nous nous étions fixés en donnant une méthode systématique de synthèse de fonctions logiques à l'aide d'un réseau cellulaire particulier.

De plus, ces études débouchent comme nous l'avons vu sur la mise en oeuvre de modules logiques universels dont l'intérêt devient important avec l'évolution des techniques de fabrication des circuits intégrés.

CHAPITRE III

ETUDE DE RESEAUX CELLULAIRES A L'AIDE DU SYSTEME ASSOCIE D'INEGALITES LINEAIRES

PREMIERE PARTIE

FORMULATION DU PROBLEME

III-1-1 - RAPPELS SUR LES FONCTIONS A SEUIL [22]

III-1-1-1 - Définitions

Une fonction logique $F(x_1, \ldots, x_n)$ des n variables binaires x_i, $i = 1, \ldots, n$ est une fonction à seuil s'il existe n + 1 nombres réels $w_1, w_2, \ldots, w_n, T$ qui satisfont le système de relations conditionnelles :

I

$$\text{a)} \quad \sum_{i=1}^{n} w_i x_i \geqslant T \qquad \text{si} \quad F(x_1, \ldots, x_n) = 1$$

$$\text{b)} \quad \sum_{i=1}^{n} w_i x_i < T \qquad \text{si} \quad F(x_1, \ldots, x_n) = 0$$

w_i, $i = 1, \ldots, n$ et T sont respectivement les poids et le seuil de la fonction logique $F(x_1, \ldots, x_n)$

III-1-1-2 - Normalisation du système I

Le système I présente l'inconvénient de comporter des inégalités au sens strict (type Ia)) et des inégalités au sens large (type Ib)) ce qui complique sa résolution. La normalisation du système I se fait de façon simple. En effet on montre [35] que les inégalités b) du système I peuvent être remplacées par :

$$\sum_{i=1}^{n} w_i x_i \leqslant T - 1 \qquad \text{si} \quad F = 0$$

Le système I est alors équivalent au système de relations conditionnelles suivant :

II

$$\text{a)} \quad \sum_{i=1}^{n} w_i x_i - T \geqslant 0 \qquad \text{si} \quad F = 1$$

$$\text{b)} \quad \sum_{i=1}^{n} w_i x_i - T \leqslant -1 \qquad \text{si} \quad F = 0$$

Posant $w_0 = -T$ le système II est équivalent à :

III

$$\text{a)} \quad \sum_{i=0}^{n} w_i x_i \geqslant 0 \qquad \text{si} \quad F = 1$$

$$\text{b)} \quad \sum_{i=0}^{n} w_i x_i \leqslant -1 \qquad \text{si} \quad F = 0$$

avec $x_0 = 1$

III-1-1-3 - Différentes représentations d'une fonction à seuil [24]

Une fonction logique à seuil peut être représentée par un système d'inégalités conditionnelles comme nous venons de le voir. Elle peut être aussi sous la forme d'une fonction booléenne classique $F(x_1, \ldots, x_n)$, c'est-à-dire par exemple une somme d'implicants premiers. Nous insisterons particulièrement dans ce paragraphe sur une troisième forme de représentation que nous utiliserons dans un paragraphe ultérieur.

Soit $y = F(x_1, x_2, \ldots, x_n)$ une fonction logique à seuil

Définition : La fonction de séparation de $F(x_1, \ldots, x_n)$ est définie par :

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n} w_i \, x_i$$

où les w_i sont les poids de F.

On écrira :

(1) $$y = F(x_1, \ldots, x_n) = \langle f(x_1, \ldots, x_n) \rangle_T$$

où T est le seuil

Exemple

$$y = x_1 + x_2 \, x_3 = \langle 2 x_1 + x_2 + x_3 \rangle_{1,5}$$

Interprétation graphique de la fonction de séparation

La relation (1) doit être vérifiée pour chaque sommet p c'est-à-dire

(2) $$y = F(t) = \langle f(p) \rangle_T$$

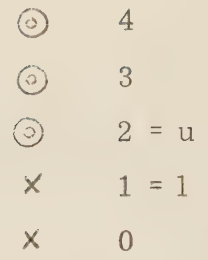
Donc à chaque sommetp où F est définie est associé un ordonné $[p, f(p), F(p)]$ où $f(p)$ est un nombre réel et $F(p) = 0$ ou 1. L'ensemble des triplets ordonnés $\{[p, f(p), F(p)]\}$ est appelé graphe de F.

Exemple

$$F(x_1, x_2, x_3) = x_1 + x_2 \, x_3 \quad \text{et} \quad f(x_1, x_2, x_3) = 2 x_1 + x_2 + x_3$$

x_1	x_2	x_3	f (p)	F (p)
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	2	1
1	0	0	2	1
1	0	1	3	1
1	1	0	3	1
1	1	1	4	1

Table de F et f



Graphe de F

◎ correspond à des sommets vrais
F (p) = 1

× correspond à des sommets faux
F (p) = 0

Figure III-1

Le graphe représente une projection particulière des sommets du cube unité sur une ligne réelle. Il est divisé en deux parties : partie 0 et partie unité. Soit u le plus petit f (p) dans la partie unité et l le plus grand f (p) dans la partie zéro. Le graphe est séparé si $u > l$.

Théorème 1 : Une fonction $F(x_1, \ldots, x_n)$ est à seuil si et seulement si il existe une fonction arithmétique linéaire de la forme :

$$f(x_1, \ldots, x_n) = \sum_{i=1}^{n} w_i x_i$$

qui conduise à un graphe séparé pour F.

Le système I est alors équivalent à :

IV

a) $f(p) \geqslant u$ si $F = 1$

b) $f(p) \leqslant l$ si $F = 0$

La relation $y = \langle f(p) \rangle_{u:l}$ représentera une forme condensée de l'écriture du système IV.

III-1-1-4 - Propriétés des fonctions à seuil

Nous énoncerons ici sans démonstration plusieurs théorèmes que nous utiliserons pour étudier les propriétés de certains réseaux cellulaires.

Théorème II : Soit

$$y_0 = F_0(y_2, y_3, \ldots, y_m) = \langle f_0(y_2, y_3, \ldots, y_m) \rangle_{u_0 : l_0}$$

et soient σ_{p0} et σ_{n0} les valeurs maximum et minimum possibles de $f_0(y_2, y_3, \ldots, y_m)$

A Si $y = y_1 + y_0$ alors $y = \langle a_1 y_1 + f_0(y_2, \ldots, y_m) \rangle_{u_0 : l_0}$

pour tout $a_1 \geqslant u_0 - \sigma_{n0}$

B si $y = y_1 . y_0$ alors $y = \langle a_1 y_1 + f_0(y_2, \ldots, y_m) \rangle_{u_0 + a_1 : l_0 + a_1}$

pour tout $a_1 \geqslant \sigma_{p0} - l_0$

Exemples

$$y = x_1 + x_2\, x_3 \ldots x_n = \langle (n-1)\, x_1 + x_2 + x_3 + \ldots + x_n \rangle_{(n-1) \,:\, (n-2)}$$

$$y = x_1\, (x_2 + x_3 + \ldots + x_n) = \langle (n-1)\, x_1 + x_2 + \ldots + x_n \rangle_{n \,:\, n-1}$$

Théorème III : Soit

$$y_0 = F_0\, (y_3,\, y_4, \ldots,\, y_m) = \langle f_0\, (y_3,\, y_4,\, \ldots,\, y_m) \rangle_{u_0 \,:\, l_0}$$

et soient σ_{p0} et σ_{n0} les valeurs maximum et minimum possibles de $f_0\, (y_3,\, y_4,\, \ldots,\, y_m)$

Si $y = y_1\, y_2 + (y_1 + y_2)\, y_0$ alors

$$y = \langle a y_1 + a y_2 + f_0\, (y_3 + y_4,\, \ldots,\, y_m) \rangle_{(u_0 + a) \,:\, l_0 + a}$$

pour tout $a \geqslant \max \left\{ (u_0 - \sigma_{n0}),\ (\sigma_{p0} - 1) \right\}$

Exemple

$$y = x_1\, x_2 + x_1\, x_3 + x_2\, x_3 + x_1\, x_4\, x_5 + x_2\, x_4\, x_5$$

$$= x_1\, x_2 + (x_1 + x_2)\, y_0 \qquad \text{avec} \qquad y_0 = x_3 + x_4\, x_5$$

D'après le théorème I $y_0 = \langle 2x_3 + x_4 + x_5 \rangle_{2:1}$

D'après le théorème II on doit choisir a de telle sorte que :

$$a \geqslant \max \left\{ (u_0 - \sigma_{n0}),\ (\sigma_{p0} - 1_0) \right\} = \max \left\{ (2 - 0),\ (4 - 1) \right\} = 3$$

Théorème IV

Si $y = \langle a_1 y_1 + a_2 y_2 + \ldots + a_m y_m \rangle_{u:1}$ alors tout y_j, $j=1, 2, \ldots, m$ peut être remplacé par $(1-y_j)$ et inversement. Le résultat est une représentation équivalente de y ayant un graphe équivalent au graphe donné.

Théorème V

Soit $y = \langle \sum_{j=1}^{m} a_j y_j \rangle_{u:1}$ alors

A $y = \langle \sum_{j=1}^{n} b\, a_j y_j \rangle_{bu:bl}$ pour tout réel $b > 0$

B $y = \langle b + \sum_{j=1}^{n} a_j y_j \rangle_{u+b:1+b}$ pour tout réel b

Conséquences des théorèmes IV et V : changement du signe des poids dans la représentation d'une fonction de séparation.

Procédure : Soit $y = \langle \sum_{j=1}^{m} a_j y_j \rangle_{u:1}$

1) pour chaque $a_j < 0$ remplacer $a_j y_j$ par $a_j - a_j \overline{y}_j$ (théorème IV)

2) Enlever tous les coefficients constants générés par 1) et remplacer $u : l$ par $(u - \sigma_n) : (l - \sigma_n)$ (théorèmes II et IV) où σ_n est la somme des coefficients négatifs.

Exemple

$$y = \langle -2x_1 + x_2 - x_3 \rangle_{-1 : -2} = \langle -2(1-\overline{x}_1) + x_2 - (1-\overline{x}_3) \rangle_{-1 : -2}$$

$$= \langle -3 + 2\overline{x}_1 + x_2 + \overline{x}_3 \rangle_{-1 : -2} = \langle 2\overline{x}_1 + x_2 + \overline{x}_3 \rangle_{2 : 1}$$

Théorème VI : Si

$$y = F(y_1, y_2, \ldots, y_m) = \langle f(y_1, y_2, \ldots, y_m) \rangle_{u:l} = \langle \sum_{j=1}^{m} a_j y_j \rangle_{u : l}$$ alors

A , $$\overline{y} = \overline{F}(y_1, y_2, \ldots, y_m) = \langle -f(y_1, y_2, \ldots, y_m) \rangle_{-l:u} = \langle \sum_{j=1}^{m} -a_j y_j \rangle_{-l:-u}$$

B , $$\overline{y} = \overline{F}(y_1, y_2, \ldots, y_m) = \langle f(\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_m) \rangle_{(\sigma-l):(\sigma-u)}$$

$$= \langle \sum_{j=1}^{m} a_j \overline{y}_j \rangle_{(\sigma-l) : (\sigma-u)}$$

où $\sigma = \sum_{j=1}^{m} a_j$

III-1-2 - DESCRIPTION D'UN RESEAU CELLULAIRE PAR UN SYSTEME D'INEGALITES LINEAIRES

Considérons la structure régulière itérative représentée sur la figure III-2.

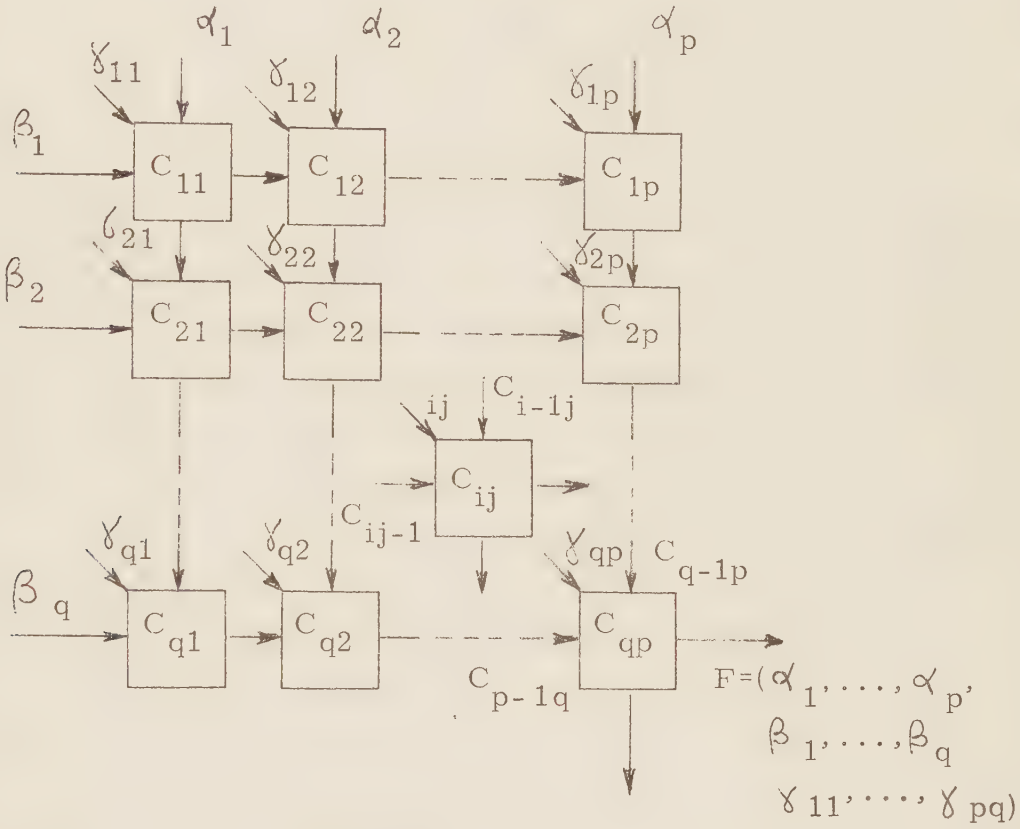


Figure III-2

Les paramètres qui définissent les dimensions du réseau sont :

- le nombre de cellules p x q
- la largeur du réseau p (p cellules par ligne)
- la hauteur du réseau q (q cellules par colonnes)
- le nombre d'entrées externes C = p + q + pq

Nous allons montrer que l'on peut associer à tout réseau du type décrit ci-dessus un système d'inégalités linéaires qui contient implicitement toutes les informations nécessaires pour l'utilisation du réseau dans la synthèse de fonctions logiques quelconques. Les dimensions du système d'inégalités associé dépendront d'une part des dimensions propres du réseau d'autre part de la complexité de la cellule de base.

III-1-2-1 - Système d'inégalités associé à la cellule de base [25]

La cellule de base comporte trois entrées α , β , γ et deux sorties qui réalisent la même fonction des variables d'entrée. Ce dédoublement de la sortie est imposé par la structure du réseau. Nous supposerons que la fonction F (α , β , γ) de sortie de la cellule est une fonction à seuil. Cette hypothèse simplificatrice n'enlève aucune généralité au problème. En effet dans le cas où la fonction F (α , β , γ) ne serait pas linéairement séparable on peut toujours la décomposer [23] en fonctions à seuil et donc se retrouver dans la configuration initiale.

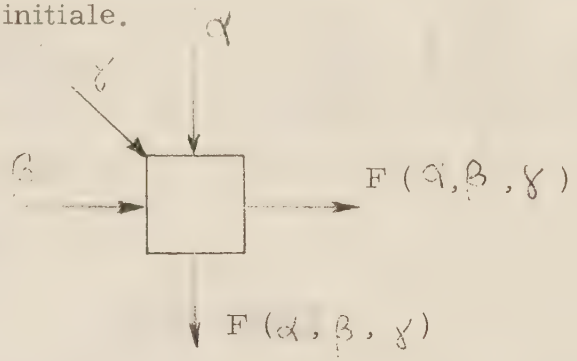


Figure III-3

Soient w_α , w_β , et w_γ les poids de la fonction à seuil F (α , β , γ) et T son seuil. Le fonctionnement de la cellule de base est alors complètement décrit par le système de relations conditionnelles :

$$V \quad \begin{cases} a) & w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \geq T \quad \text{si} \quad F^{(j)}(\alpha, \beta, \gamma) = 1 \\ b) & w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \leq T-1 \quad \text{si} \quad F^{(j)}(\alpha, \beta, \gamma) = 0 \end{cases}$$

où $\alpha^{(j)}$, $\beta^{(j)}$, $\gamma^{(j)}$ et $F^{j}(\alpha, \beta, \gamma)$ représentent les valeurs respectives des variables d'entrée α, β, γ et de la fonction $F(\alpha, \beta, \gamma)$ qui correspondent à une combinaison des variables d'entrée dont l'équivalent décimal est j (Table II)

j	α	β	γ
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Table II

Posant $w_0 = -T$ le système V devient :

$$VI \quad \begin{cases} a) & w_0 + w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \geq 0 \quad \text{si} \quad F^{(j)}(\alpha, \beta, \gamma) = 1 \\ b) & w_0 + w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \leq -1 \quad \text{si} \quad F^{(j)}(\alpha, \beta, \gamma) = 0 \end{cases}$$

III-1-2-2 - Passage d'un système de relations conditionnelles à un système d'inégalités linéaires

On peut convertir le système VI de relations conditionnelles en un ensemble d'inégalités linéaires

$$\text{VII} \quad \begin{cases} \text{a)} & w_0 + w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \geq -U \times [1 - F^{(j)}(\alpha, \beta, \gamma)] \\ \text{b)} & w_0 + w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \leq -1 + U F^{(j)}(\alpha, \beta, \gamma) \end{cases}$$

En effet supposons que U soit un entier positif suffisamment grand. Alors, si $F^{(j)}(\alpha, \beta, \gamma) = 1$, VII a) est équivalent à :

$$w_0 + w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \geq 0$$

et VII b) est équivalent à :

$$w_0 + w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \leq -1 + U$$

qui est toujours vérifiée

si $F^{j}(\alpha, \beta, \gamma) = 0$ VII a) est équivalent à :

$$w_0 + w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \geq -U$$

qui est toujours vérifiée

et VII b) est équivalent à

$$w_0 + w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \leq -1$$

Donc le système d'inégalités linéaires VII est bien équivalent au système de relations conditionnelles VI pour un choix convenable de l'entier positif U.

Remarque sur le choix de U

U est suffisamment grand s'il n'est pas plus petit que le maximum concevable de la valeur absolue du côté gauche de chaque inégalité du système VII. Pour VII a) par exemple un entier U qui vérifierait

$$U \geq \left| w_0 + w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \right|$$

$$= \left| w_0 \right| + \left| w_\alpha \right| + \left| w_\beta \right| + \left| w_\gamma \right|$$

serait un choix convenable

On connaît [26] une borne supérieure pour les poids w_i et le seuil T d'un élément à seuil à d entrées :

$$(3) \qquad \left| w_i \right| \leq 2 \left(\frac{d+1}{4}\right)^{\frac{d+1}{2}}$$

$$(4) \qquad \left| T \right| \leq 2 d \left(\frac{d+1}{4}\right)^{\frac{d+1}{2}}$$

Les relations (3) et (4) permettent donc de trouver dans tous les cas une valeur de U rendant compatible le système VII. Nous verrons que cette valeur ne représente pas une borne inférieure de U.

III-1-2-3 - Système d'inégalités linéaires associé à un réseau cellulaire

Nous venons de voir que le système associé à un opérateur à seuil pouvait se mettre sous la forme

VIII
$$w_0 + w_\alpha \alpha^{(j)} + w_\beta \beta^{(j)} + w_\gamma \gamma^{(j)} \geq -U[1-F^{(j)}(\alpha, \beta, \gamma)]$$
$$-w_0 - w_\alpha \alpha^{(j)} - w_\beta \beta^{(j)} - w_\gamma \gamma^{(j)} \geq 1-U F^{(j)}(\alpha, \beta, \gamma)$$

Donc à chaque cellule du réseau de la figure III-2 nous pouvons associer un système de ce type. Nous supposerons que toutes les cellules du réseau sont identiques, c'est-à-dire qu'elles réalisent des fonctions à poids et seuil identiques. Les entrées des cellules seront soit uniquement des entrées externes telles que $\alpha_1, \ldots, \alpha_p, \ldots, \beta_1, \ldots, \beta_q, \gamma_{11}, \ldots, \gamma_{pq}$ soit des entrées externes et des sorties de cellules voisines. Pour simplifier l'écriture nous noterons du même symbole, la cellule et la fonction réalisée par la cellule.

Ainsi pour le réseau de la figure III-2 le système d'inégalités associé est le suivant :

$$\left\{\begin{array}{l}
C_{11}\left[\begin{array}{l} w_0+w_\alpha \alpha_1^{(j)}+w_\beta \beta_1^{(j)}+w_\gamma \gamma_{11}^{(j)} \geq -U(1-C_{11}^{(j)}(\alpha_1, \beta_1, \gamma_{11})) \\ -w_0-w_\alpha \alpha_1^{(j)}-w_\beta \beta_1^{(j)}-w_\gamma \gamma_{11}^{(j)} \geq 1-U C_{11}^{(j)}(\alpha_1, \beta_1, \gamma_{11}) \end{array}\right. \\
C_{12}\left[\begin{array}{l} w_0+w_\alpha \alpha_2^{(j)}+w_\gamma \gamma_{12}^{(j)}+w_\beta C_{11}^{(j)} \geq -U(1-C_{12}^{(j)}(\alpha_2, C_{11}, \gamma_{12})) \\ -w_0-w_\alpha \alpha_2^{(j)}-w_\gamma \gamma_{12}^{(j)}-w_\beta C_{11}^{(j)} \geq 1-U C_{12}^{(j)}(\alpha_2, C_{11}, \gamma_{12}) \end{array}\right. \\
\hrulefill \\
C_{ij}\left[\begin{array}{l} w_0+w_\alpha C_{i-1j}^{(j)}+w_\gamma \gamma_{ij}^{(j)}+w_\beta C_{ij-1}^{(j)} \geq -U(1-C_{ij}^{(j)}(C_{i-1j}, \gamma_{ij}, C_{ij-1})) \\ -w_0-w_\alpha C_{i-1j}^{(j)}-w_\gamma \gamma_{ij}^{(j)}-w_\beta C_{ij-1}^{(j)} \geq 1-U C_{ij}^{(j)}(C_{i-1j}, \gamma_{ij}, C_{ij-1}) \end{array}\right. \\
\hrulefill
\end{array}\right.$$

$$C_{pq}\left[\begin{array}{l} w_0+w_\alpha C_{q-1p}^{(j)}+w_\gamma \gamma_{pq}^{(j)}+w_\beta C_{p-1q}^{(j)} \geq -U(1-C_{pq}^{(j)}(C_{q-1p}, \gamma_{pq}, C_{p-1q})) \\ -w_0-w_\alpha C_{q-1p}^{(j)}-w_\gamma \gamma_{pq}^{(j)}-w_\beta C_{p1-q}^{(j)} \geq 1-U C_{pq}^{(j)}(C_{q-1p}, \gamma_{pq}, C_{p-1q}) \end{array}\right.$$

Le système IX contient implicitement toutes les informations nécessaires pour résoudre le problème de la synthèse d'une fonction logique arbitraire à l'aide d'un réseau cellulaire qui peut en général s'énoncer de la façon suivante :

Etant donné un réseau cellulaire à structure d'interconnexion connue et figée composé de cellules de base réalisant une fonction donnée et une fonction logique arbitraire F déterminer les assignements des entrées externes du réseau telles que la fonction F soit réalisée à une sortie donnée du réseau.

Donc dans un problème de synthèse le système IX comportera

- des inconnues qui seront les assignements des entrées externes $\alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q$ et $\gamma_{11}, \ldots, \gamma_{pq}$ et les sorties intermédiaires des cellules.

- des données qui seront les poids et le seuil de la fonction réalisée par la cellule de base, le nombre entier positif U que l'on sait déterminer et les valeurs $C_{ij}^{(j)}$ correspondant à la cellule C_{ij} du réseau dont les sorties doivent réaliser F. En général la fonction donnée F est réalisée en sortie de la cellule C_{qp}.

Dimensions du système d'inégalités linéaires IX

Soit n le nombre de variables d'entrée, P le nombre d'inconnues et Q le nombre d'inégalités. On a :

(5) $P = p + q + pq + (pq-1)\,2^n$

(6) $Q = 2\,pq\,2^n$

Les dimensions du système d'inégalités associé à un réseau cellulaire sont donc relativement importantes. Cette caractéristique constitue une des principales limitations de la méthode de synthèse de fonctions logiques quelconques à l'aide de réseaux cellulaires que nous décrirons ultérieurement. Nous montrerons toutefois que dans certains cas il est possible de réduire dans des proportions importantes les dimensions de ce système.

III-1-2-4 - Formulation du problème de la résolution du système d'inégalités linéaires associé à un réseau cellulaire en un problème de programmation linéaire

En utilisant une notation matricielle nous pouvons mettre le système IX sous la forme :

(7) $\overline{B} + A\overline{X} \geqslant 0$

où B est un vecteur colonne de dimensions $Q = 2\,pq\,2^n$

A est une matrice rectangulaire de dimensions Q x P

X est un vecteur colonne de dimensions $P = p+q + pq + (pq-1)\,2^n$

dont les composantes ne peuvent prendre que la valeur 0 ou 1. Ces composantes sont les inconnues du problème, c'est-à-dire d'une part les assignements des entrées externes $\alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q, \gamma_{11}, \ldots, \gamma_{pq}$ dans l'ensemble des variables d'entrée choisies, d'autre part les valeurs des sorties intermédiaires pour les différentes valeurs de j.

Un problème de programmation linéaire en variables bivalentes s'énonce généralement de la façon suivante [27] :

(8) Minimizer $\overline{C} \cdot \overline{X}$

$\overline{C}$ étant un vecteur coût de dimension n (composantes non négatives)

Le vecteur $\overline{X}$ étant soumis à l'ensemble des contraintes :

(9) $\overline{B} + A\overline{X} \geqslant 0$

$\overline{B}$ étant un vecteur colonne de dimensions m, A une matrice de dimensions m x n et $\overline{X}$ un vecteur colonne de dimensions n dont les composantes sont 0 ou 1.

Nous appelerons :

- solution : tout vecteur $\overline{X}$, de coordonnées 0 ou 1

- solution réalisable : toute solution qui satisfait le système de contraintes $\overline{B} + A\overline{X} \geqslant 0$

- solution réalisable optimale : une solution réalisable qui minimize $\overline{C} \cdot \overline{X}$.

Nous voyons donc que la résolution du système d'inégalités (7) associé au réseau cellulaire se ramène à la recherche d'une solution réalisable d'un problème de programmation linéaire en variables bivalentes.

Donc la synthèse d'une fonction logique arbitraire à l'aide d'une structure cellulaire composée d'opérateurs à seuil passera par la recherche d'une solution réalisable d'un problème de programmation linéaire. Nous décrirons dans le paragraphe suivant les méthodes et algorithmes que nous avons utilisés.

III-1-2-5 - Méthodes et algorithmes de programmation linéaire en variables bivalentes

La méthode la plus "simpliste" pour résoudre le problème de programmation linéaire en variables bivalentes :

$$\left.\begin{array}{ll} (8) & \text{Minimiser } \overline{C} \,.\, \overline{X} \\ (9) & \text{soumis à } \overline{B} + A\,\overline{X} \geqslant 0 \end{array}\right\} \quad P$$

$\overline{X}$ vecteur colonne de dimensions n
et de composantes $x_j = 0$ ou 1

consiste en l'énumération pure et simple de toutes les solutions. En effet comme les variables de (P) ne peuvent prendre que les valeurs 0 et 1, l'ensemble $U = \{ u \}$ de toutes les solutions de (P) est fini et le nombre d'éléments dans U est 2^n, où n est le nombre de variables x_j. La procédure d'énumération et d'essai des 2^n combinaisons possibles (c'est-à-dire des solutions) est illustrée pour un problème à cinq variables sur la figure III-4. Cette figure présente cinq niveaux. Partant du sommet A, à chaque niveau nous assignons tour à tour les valeurs 0 et 1 à la variable ayant le numéro du niveau. Tous les points le long de la ligne AB représentent la même solution, tandis que tous les points le long de la ligne AG représentent des solutions différentes. En continuant cette procédure jusqu'à ce que toutes les variables aient été assignées aux valeurs 0 et 1, nous obtenons l'ensemble des $2^5 = 32$ solutions. L'emploi de cette méthode n'est évidemment concevable que pour des problèmes de dimensions très réduites ($n \leqslant 4$ environ).

Dans un article de synthèse sur la programmation linéaire entière M. L. BALINSKI [28] émettait l'avis que des schémas "habiles" d'énumération sont dans certaines situations, au moins, les meilleures méthodes connues de solution.

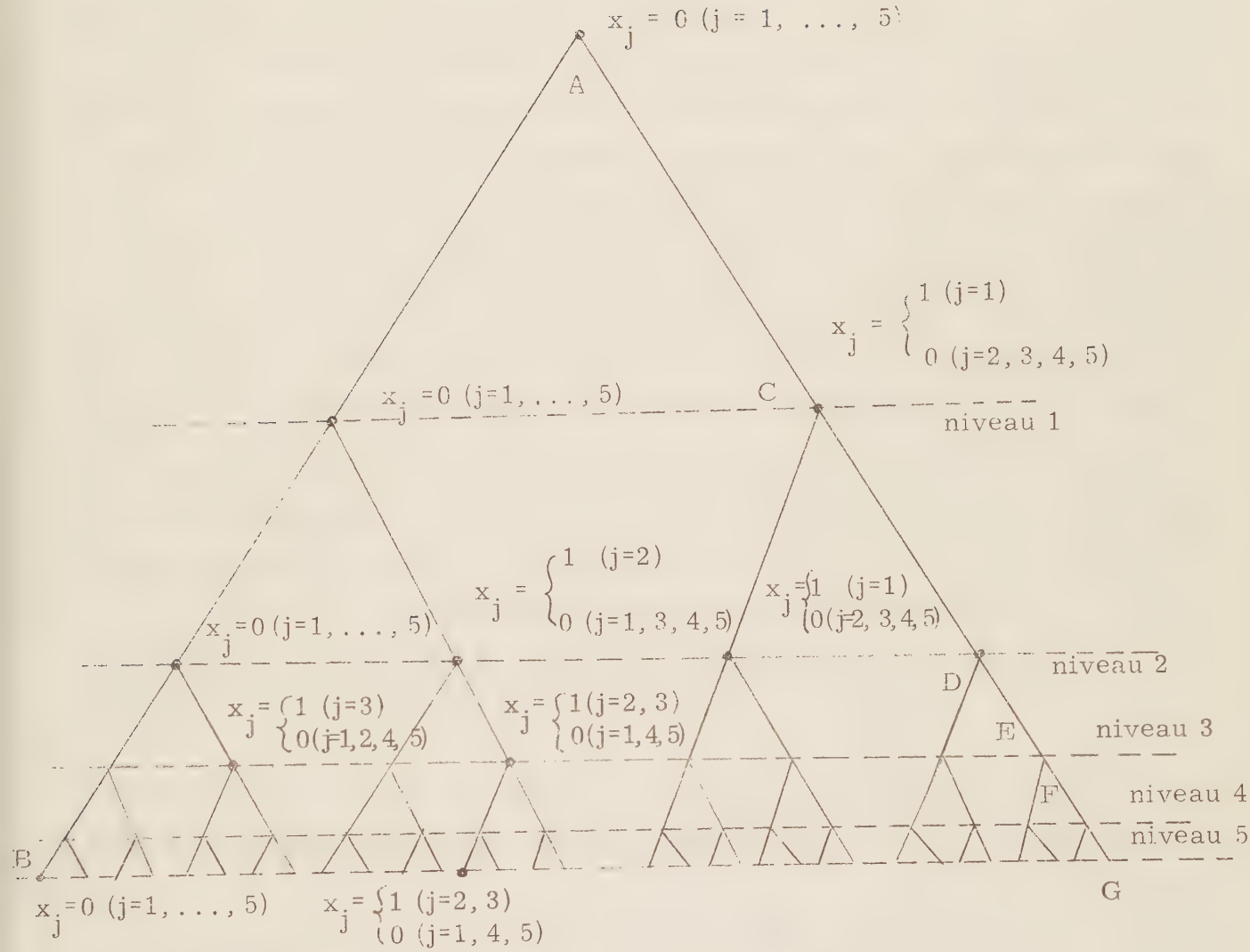


Figure III-4

Le qualificatif "habile" se rapportait sans aucun doute au fait que l'on devait faire mieux que l'énumération complète. Ceci est rendu possible par l'emploi de stratégies qui conduisent, en même temps que l'énumération avance à la génération d'informations que l'on peut employer pour exclure un

grand nombre de solutions de toute considération ultérieure. L'exclusion de solutions qui suggère le terme descriptif d'énumération implicite pour de telles méthodes peut être accomplie de nombreuses manières, par exemple en utilisant la présence d'une borne de la valeur optimale de la fonction objectif qui s'améliore progressivement au fur et à mesure que des solutions réalisables meilleures sont trouvées. Il est dommage que l'exclusion de solutions puisse quelquefois donner naissance à la nécessité de mettre en mémoire un très grand nombre d'informations afin de se rappeler quelles solutions ont été exclues ou bien énumérées. Donc un "bon" algorithme doit combiner un "taux d'exclusion" suffisamment élevé avec des besoins de mise en mémoire suffisamment modestes. Nous touchons là la caractéristique essentielle de la résolution d'un problème de programmation linéaire en variables bivalentes.

Un support additionnel au point de vue de Balinski a été fourni par la publication de l'algorithme additif de Balas [27] et de quelques extensions utiles proposées par Glover [29]. L'algorithme de Balas permet en particulier de simplifier au maximum des opérations de "pivotage".

Parmi les avantages de l'énumération implicite en général et de l'approche de Balas en particulier nous citerons :

a) l'addition est la seule opération arithmétique requise (pas de résolutions de système d'équations linéaires), ce qui élimine les problèmes d'erreurs d'arrondi.

b) une solution réalisable, que l'on espère bonne, se trouve usuellement en mémoire si les calculs sont arrêtés avant la terminaison naturelle par exhaustion.

c) il est aisé de contrôler le taux d'énumération implicite à mesure que les calculs se développent ce qui rend aisée la mise en place de règles d'arrêt.

Ces avantages nous ont conduit à utiliser des algorithmes s'inspirant des caractéristiques signalées ci-dessus. Nous décrivons dans la suite de ce paragraphe les deux algorithmes qui ont été développés au L.A.A.S. avec l'aide du Service Informatique et Simulation.

III-1-2-5-1 - Principe de l'énumération implicite [30]

L'ensemble de toutes les solutions étant fini, nous pouvons le représenter par un arbre. Nous avons représenté cet arbre pour n = 5 sur la figure III-4.

Une solution partielle S est définie comme un assignement de valeurs binaires à un sous-ensemble de n variables. Toute variable à laquelle n'est assignée aucune valeur est libre. Nous adopterons la convention de notation suivante : le symbole j dénotera xj = 1 et le symbole - j dénotera xj = 0. Donc, si n = 5 et S = $\{ 3, 5, -2 \}$, alors $x_3 = 1$, $x_5 = 1$, $x_2 = 0$ et les variables x_1 et x_4 sont libres. Une complétion d'une solution partielle S est définie comme une solution étant déterminée par S plus l'ensemble des valeurs pouvant être prises par les variables libres. Dans l'exemple ci-dessus il y a quatre complétions possible de S :

0	0	1	0	1
0	0	1	1	1
1	0	1	0	1
1	0	1	1	1

Donc une solution partielle formée de s éléments détermine un ensemble de 2^{n-s} complétions différentes. Quand il n'y a aucune variable libre, il n'existe qu'une seule complétion de S : S lui-même.

L'énumération implicite comprend la génération d'une séquence de solutions partielles, et l'examen simultané de toutes les complétions de chacune d'elles. En d'autres termes l'énumération implicite fournit un ensemble de règles selon lesquelles on peut obtenir une solution optimale en ne suivant que certaines branches de l'arbre. Partant d'une situation dans laquelle

les n variables valent 0, la méthode consiste à assigner la valeur 1 à quelques variables de sorte qu'après avoir essayé une petite partie des 2^n combinaisons, on obtienne soit une solution optimale soit l'assurance qu'il n'existe pas de solution réalisable.

Au cours de la recherche d'une solution réalisable, on explorera l'arbre en suivant certaines branches. Il se peut que l'on arrive sur un noeud tel qu'aucune branche qui en sont issues ne mène à une solution réalisable. On remontera alors les dernières arêtes parcourues jusqu'à un noeud tel que la seconde des branches qui en partent n'ait pas encore été explorée et puisse mener à priori à une solution réalisable.

Dans les programmes, ce cheminement sur l'arbre sera représenté par une pile simple S. Chaque branche de l'arbre sera représentée par deux mots :

- dans le premier mot, la valeur de l'indice de la variable correspondant à la branche

si la variable est assignée à 1, le contenu du mot est égal à l'indice
si la variable est assignée à 0, le contenu du mot est égal à la valeur opposée de l'indice.

- dans le second mot, un indicateur permettant de savoir si la seconde branche issue du noeud considéré a déjà été parcourue

si oui, la valeur de l'indicateur est 2 (on dit alors que la variable est soulignée)
si non, la valeur de l'indicateur est 0.

Ainsi si l'on arrive à un noeud qui ne fournit que des solutions non réalisables, on recherchera dans la pile le dernier élément dont l'indicateur soit égal à 0. On changera alors le signe de l'indice dans le premier mot, on donnera à l'indicateur la valeur 2, et on effacera tous les éléments qui suivent celui-ci dans la pile. On pourra alors recommencer l'exploration de l'arbre des solutions.

III-1-2-5-2 - Premier algorithme de programmation linéaire en variables bivalentes

Le système est défini à chaque instant par quatre vecteurs :

S représentant la solution

X représentant les valeurs des variables

T représentant l'ensemble des indices des variables contenues dans la solution

Y représentant le vecteur des écarts et défini par

$$y_i = b_i + \sum_{j \in T} a_{ij} x_j \qquad i \in I$$

Les différents pas de l'algorithme sont les suivants :

1. Recherche des variables candidates

On forme l'ensemble des indices J_c défini par :

$$j \in J_c \quad \text{si} \quad \exists_i \quad \text{tel que} \quad y_i < 0 \quad \text{et} \quad a_{ij} > 0$$

de telle sorte que l'un au moins des écarts y_i augmente si la variable j est incluse dans la solution.

2. Test d'impossibilité

Si pour $j \in J_c$ $\exists_i \in I$ tel que :

$$\left[y_i + \sum_{\substack{j \in J_c \\ a_{ij} > 0}} a_{ij} \right] < 0$$

alors la solution partielle considèrée ne pourra jamais fournir de solution réalisable. On se branche alors sur la procédure de retour sur la solution.

3. Choix d'une nouvelle variable

On choisit comme nouvelle variable dans la solution, $j_0 \in J_c$ qui maximise l'expression

$$\sum_{i \in I} \left[\min (0, y_i + a_{ij}) \right]$$

Ce test correspond au choix de la variable libre qui augmente globalement Y de manière optimale. La nouvelle variable x_{j0} se voit affecter la valeur 1 et entre dans la solution. On retourne alors au pas 1 de l'algorithme.

4. Retour sur la solution

Il s'agit, comme nous l'avons vu, de remonter dans l'arbre des solutions et de reprendre le dernier chemin laissé inexploré.

S'il en existe un, on réinitialise Y et X et l'on repart au pas 1.

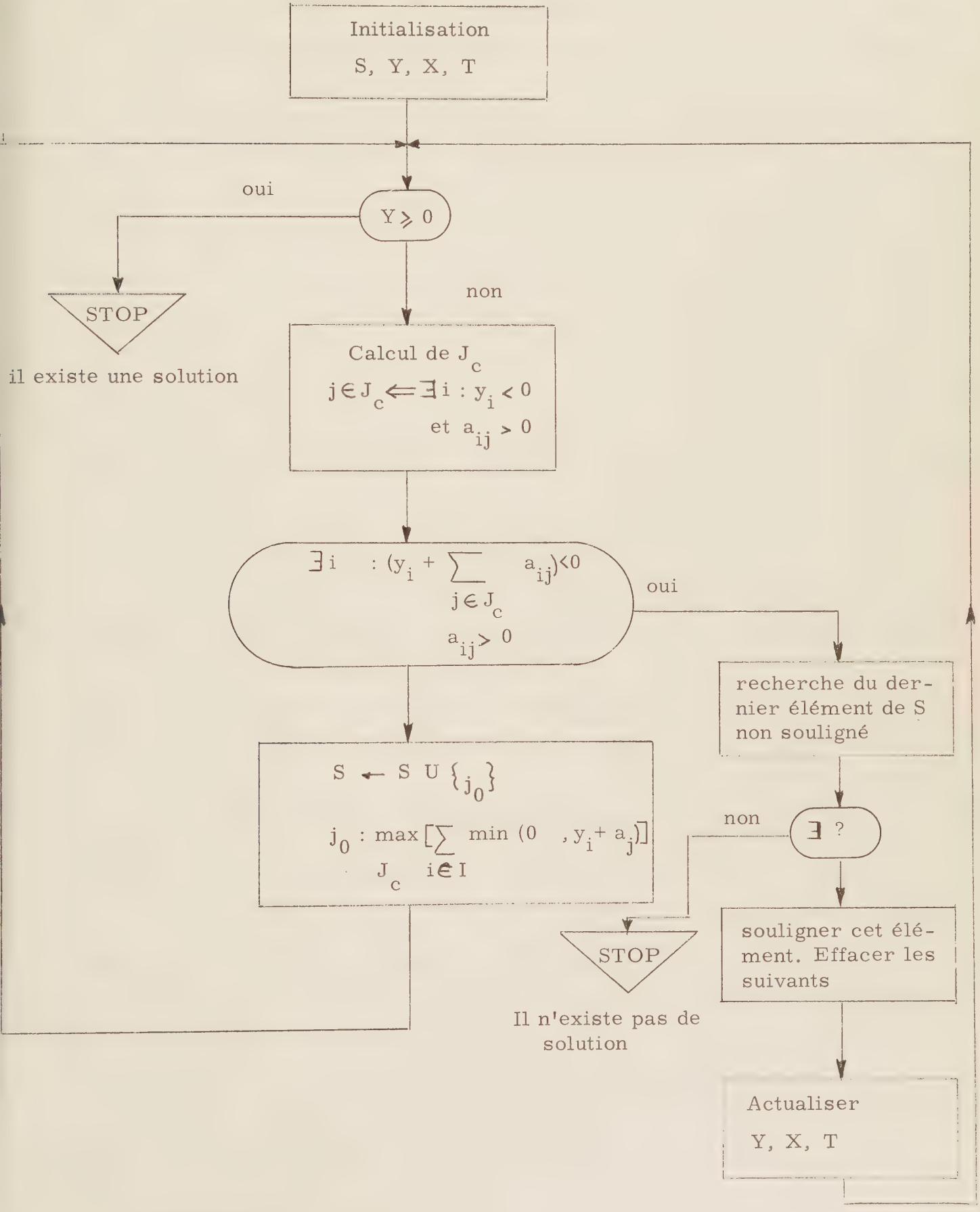
S'il n'en existe pas, le problème n'a pas de solution et le programme s'achève.

III-1-2-5-3 - Second algorithme de programmation linéaire en variables bivalentes [31]

Le principe est le même que celui de l'algorithme précédent mais il existe une différence notable lors de l'exploration de l'arbre des solutions.

Chaque fois que nous choisissons une nouvelle variable (c'est-à-dire que l'on parcourt une arête sur l'arbre des solutions) nous essaierons de

Organigramme du premier algorithme (voir Annexe I pour le listing Fortran)



déterminer des variables libres que l'on doit forcer à certaines valeurs de manière à éviter un certain nombre de solutions impossibles.

Pour contrôler l'évolution du programme nous définissons pour toute solution partielle S trois vecteurs colonnes de dimensions m + 1 :

$Y = \{ y_i \}_{i \in I}$ défini par $y_i = b_i + \sum_{j \in W} a_{ij} x_j$

$U = \{ u_i \}_{i \in I}$ défini par $u_i = y_i + \sum_{\substack{j \notin W \\ a_{ij} > 0}} a_{ij}$

$L = \{ l_i \}_{i \in I}$ défini par $l_i = y_i + \sum_{\substack{j \notin W \\ a_{ij} < 0}} a_{ij}$

où W représente l'ensemble des indices des variables assignées de S. En d'autres termes Y est la valeur courante de l'inégalité i sommée sur seulement les valeurs assignées de S. L donne la valeur minimum de l'inégalité i vu qu'elle représente la valeur courante plus tous les coefficients négatifs des variables libres. De la même façon U donne la valeur maximum de l'inégalité i pour la solution partielle S.

Nous utiliserons également un vecteur $T = \{ t_i \}_{i \in I}$ défini par $t_i = 1$ si l'inégalité correspondante peut fournir des indications sur des variables que l'on peut forcer, $t_i = 0$ dans le cas contraire ou si $l_i \geq 0$.

En effet si $l_i \geq 0$ pour un certain i, l'inégalité i ne peut jamais prendre une valeur négative quelle que soit la complétion de S et cette inégalité n'est donc pas contraignante. En d'autres termes l'inégalité i ne peut jamais causer un retour sur la solution ni forcer des variables à être spécifiées vu qu'elle est toujours satisfaite. Donc nous pouvons ignorer l'inégalité i jusqu'à ce que un retour en arrière ne rejette éventuellement la solution partielle présenté S.

1. Vérification cyclique des contraintes

Cette étape du programme cherche à déterminer à partir d'une solution partielle si l'on peut assigner à certaines variables libres des valeurs imposées par la solution afin d'éviter l'énumération du maximum de solutions impossibles.

Pour cela nous vérifierons les inégalités pouvant fournir une certaine information sur la solution partielle (inégalités correspondant aux $t_i = 1$). Le balayage s'effectuera de la ligne 1 à la ligne m, puis de nouveau de la ligne 1 à la ligne m jusqu'à ce que l'un des deux cas suivants se produisent.

a - il existe un élément u_i du vecteur U strictement négatif: alors l'inégalité i ne peut être satisfaite quelle que soit la complétion de la solution partielle S. La solution partielle considérée est alors impossible et il faut effectuer un retour sur la solution.

b - le vecteur T est nul, c'est-à-dire que nous n'obtiendrons pas d'informations supplémentaires et que nous ne pouvons plus forcer aucune variable libre.

Pour déterminer en cours de vérification cyclique quelles sont les variables que nous pouvons forcer, nous appliquons le test suivant :

Si il existe $j \notin W$ et $i \in I$ tels que $|a_{ij}| > u_i$ alors nous pouvons poser :

$x_j = 1$ si $a_{ij} > 0$

$x_j = 0$ si $a_{ij} < 0$

En effet reécrivons les expressions de y_i, u_i, l_i :

$$y_i^{(k)} = b_i + \sum_{j \in W_k} a_{ij} x_j$$

$$u_i^{(k)} = y_i^{(k)} + \sum_{\substack{j \in W_k \\ a_{ij} > 0}} a_{ij}$$

$$l_i^{(k)} = y_i^{(k)} + \sum_{\substack{j \notin W_k \\ a_{ij} < 0}} a_{ij}$$

1. Supposons que $|a_{ij0}| > u_i^{(k)}$ et $a_{ij0} < 0$ et imposons à x_{j0} la valeur 1 : $x_{j0} = 1$. Nous avons alors :

$$W_{k+1} = W_k + \{j_0\} \qquad y_i^{(k+1)} = y_i^{(k)} + a_{ij0}$$

$$u_i^{(k+1)} = y_i^{(k+1)} + \sum_{\substack{j \notin W_{k+1} \\ a_{ij} > 0}} a_{ij}$$

Or $\sum_{\substack{j \notin W_{k+1} \\ a_{ij} > 0}} a_{ij} = \sum_{\substack{j \notin W_k \\ a_{ij} > 0}} a_{ij}$ puisque $a_{ij0} < 0$

d'où

$$u_i^{(k+1)} = y_i^{(k)} + a_{ij0} + \sum_{\substack{j \notin W_k \\ a_{ij} > 0}} a_{ij} = u_i^{(k)} + a_{ij0} = u_i^{(k)} - |a_{ij0}|$$

Donc $u_i^{(k+1)} < 0$ et la solution S_{k+1} est impossible.

Les conditions $|a_{ij0}| > u_i^{(k)}$ et $a_{ij0} > 0$ impose donc bien $x_{j0} = 0$.

2. Supposons que $|a_{ij0}| > u_i^{(k)}$ et $a_{ij0} > 0$ et imposons $x_{j0} = 0$. Nous avons alors :

$$y_i^{(k+1)} = y_i^{(k)} + a_{ij0} \cdot x_{j0} = y_i^{(k)}$$

$$u_i^{(k+1)} = y_i^{(k+1)} + \sum_{\substack{j \notin W_{k+1} \\ a_{ij} > 0}} a_{ij}$$

or

$$\sum_{\substack{j \notin W_{k+1} \\ a_{ij} > 0}} a_{ij} = \sum_{\substack{j \notin W_k \\ a_{ij} > 0}} a_{ij} - a_{ij0}$$

d'où

$$u_i^{(k+1)} = y_i^{(k)} + \sum_{\substack{j \notin W_k \\ a_{ij} > 0}} a_{ij} - a_{ij0} = u_i^{(k)} - |a_{ij0}| < 0$$

Donc la solutions S_{k+1} est impossible et nous devons prendre $x_{j0} = 1$.

Nous voyons qu'une solution partielle S_k peut forcer un certain ensemble de variables $J_f^{(k)}$ à prendre des valeurs déterminées. Une constatation supplémentaire est qu'il n'est pas nécessaire d'explorer dans l'arbre les branches correspondant aux valeurs complémentaires prises par ces variables forcées. C'est pourquoi ces variables seront incluses dans la solution et soulignées.

La vérification cyclique se fera donc essentiellement d'après les valeurs de T. Nous avons vu que :

si $t_i = 1$ il faut vérifier la $i^{ème}$ inégalité pour s'assurer qu'il n'existe aucune variable à forcer ou que $u_i < 0$ ou $l_i \geq 0$.

si $t_i = 0$ on ne s'intéresse pas à la $i^{ème}$ inégalité.

Considérons maintenant les règles de construction et de variation du vecteur T

a. La valeur initiale est :

$t_i = 1$ si $l_i < 0$

$t_i = 0$ si $l_i \geq 0$

pour $i = 0, 1, \ldots, m$

b. Après avoir vérifié la $i^{ème}$ inégalité on pose $t_i = 0$ car cette inégalité ne pourra plus fournir d'information supplémentaire jusqu'au premier changement dans la solution.

c. Lorsqu'une variable libre x_j est spécifiée, le vecteur T évolue de la manière suivante :

c.1 - si $x_j = 0$, alors

$t_i^{(k+1)} = 1$ si $a_{ij} > 0$ et $l_i < 0$

$t_i^{(k+1)} = 0$ si $a_{ij} < 0$ et $l_i \geq 0$

$t_i^{(k+1)} = t_i^{(k)}$ dans les autres cas

c.2 - si $x_j = 1$, alors

$t_i^{(k+1)} = 1$ si $a_{ij} < 0$ et $l_i < 0$

$t_i^{(k+1)} = 0$ si $a_{ij} > 0$ et $l_i \geq 0$

$t_i^{(k+1)} = t_i^{(k)}$ dans les autres cas.

Les autres étapes de l'algorithme sont d'une part le choix de la variable à introduire dans la solution et d'autre part le retour en arrière sur la solution.

Le critère de choix d'une nouvelle variable est différent de celui du premier algorithme et se traduit par les opérations suivantes :

Pour chaque variable libre x_j soit L_j le nombre d'inégalités qui satisfont à la fois

$$l_i < 0 \quad \text{et}$$

$$l_i + a_{ij} \geq 0$$

On choisit comme nouvelle variable celle dont l'indice j_0 maximise L_j. La solution partielle suivante est alors :

$$S_{k+1} = \left\{ S_k, j_0 \right\}$$

En d'autres termes cette méthode a pour but de maximiser le nombre d'inégalités où $l_i (S_{k+1}) \geq 0$ quand la valeur 1 est assignée à x_j, de sorte que la nouvelle solution partielle possède autant d'inégalités non contraignantes que possible. Si plusieurs variables ont le même nombre d'inégalités où $l_i (S_{k+1}) \geq 0$, un second critère est appliqué qui permet de trancher parmi les j correspondant à un maximum de L_j on calcule la valeur

$$\sum_i a_{ij}$$

pour tous les i tels que

$$a_{ij} > 0$$

$$l_i + a_{ij} < 0$$

On choisit alors le j_0 qui maximise $\sum_i a_{ij}$.

III-1-2-5-4 - Conclusion

La table III donne les résultats expérimentaux obtenus à l'aide des deux algorithmes que nous venons de décrire. Nous reviendrons plus en détails sur les performances de ces algorithmes lorsque nous décrirons des exemples précis d'applications dans la deuxième partie de ce chapitre mais nous pouvons

déjà conclure de la table III que le deuxième algorithme semble beaucoup plus adapté au problème qui se pose à nous :

- plus d'inégalités que de variables et donc une région de solutions assez petites

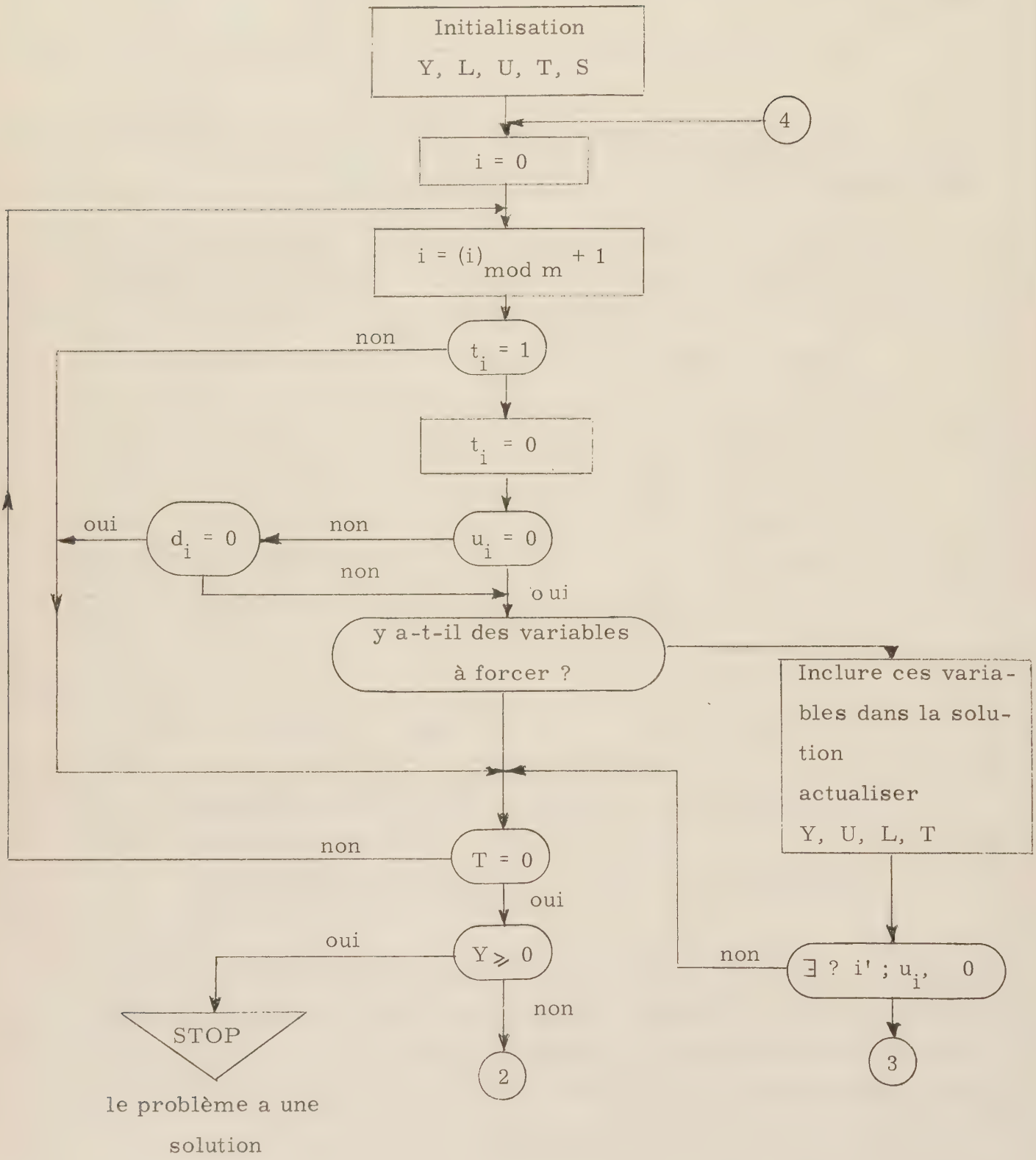
- une matrice de coefficients A telle que la majorité des éléments soient nuls et que les autres éléments soient essentiellement 1 ou - 1.

Algorithme 1 (7044)		Algorithme 2 (7044)	
Système 24 x 24		Système 24 x 24	
7 itérations	0, 8''	6 itérations	2''
7 itérations	0, 8''	9 itérations	2, 8''
9 itérations	1''	14 itérations	4, 3''
10 itérations	1''	6 itérations	2''
226 itérations	14, 9''	23 itérations	7, 1''
148 itérations	9, 7''	20 itérations	6, 2''
Système 76 x 69		Système 76 x 69	
24. 000 itérations	1 h	154 itérations	18''
		110 itérations	12''
20. 000 itérations	1 h	199 itérations	23''
		899 itérations	1'45''
		2004 itérations	4'04''
		199 itérations	22''5

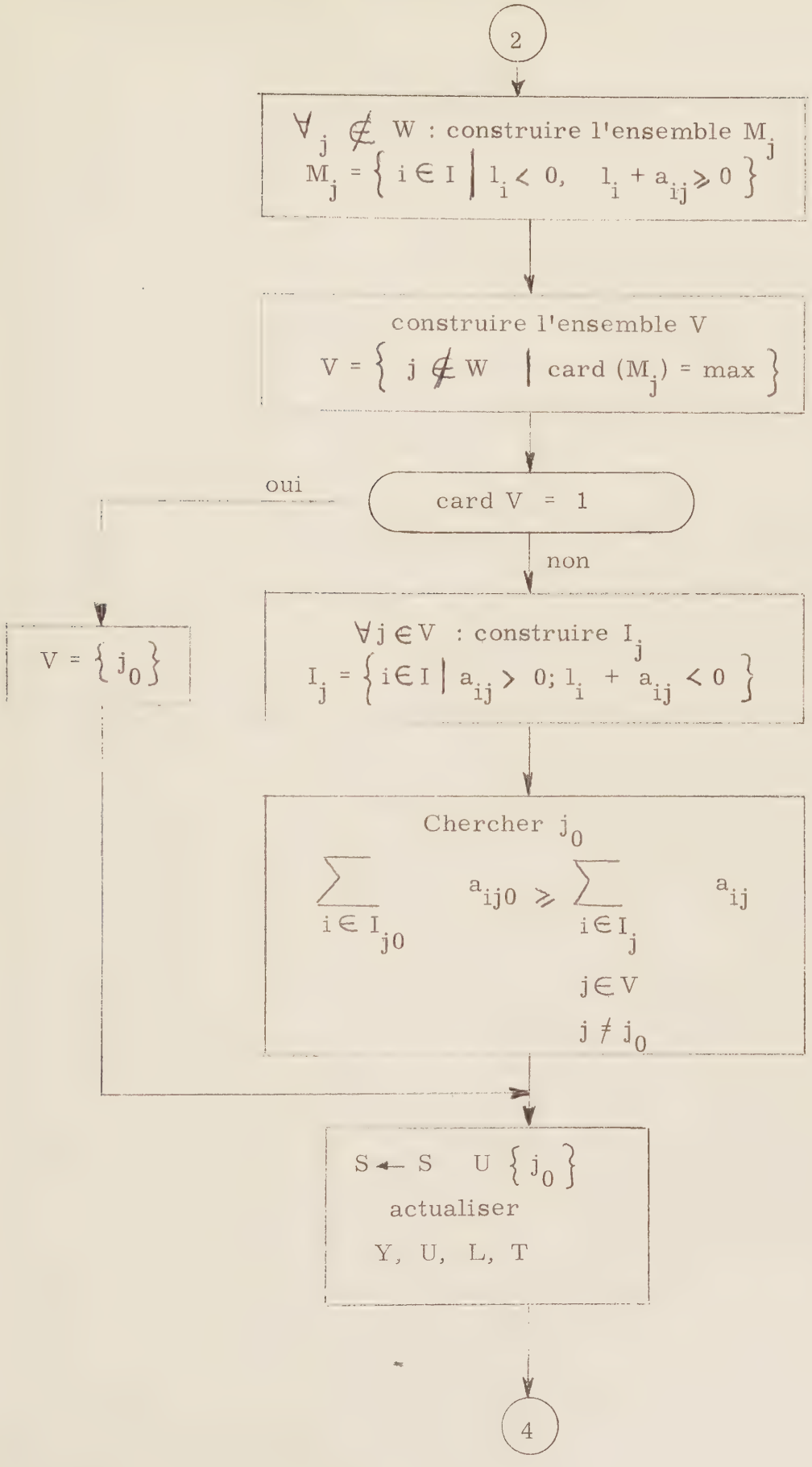
Table III

Organigramme du deuxième algorithme

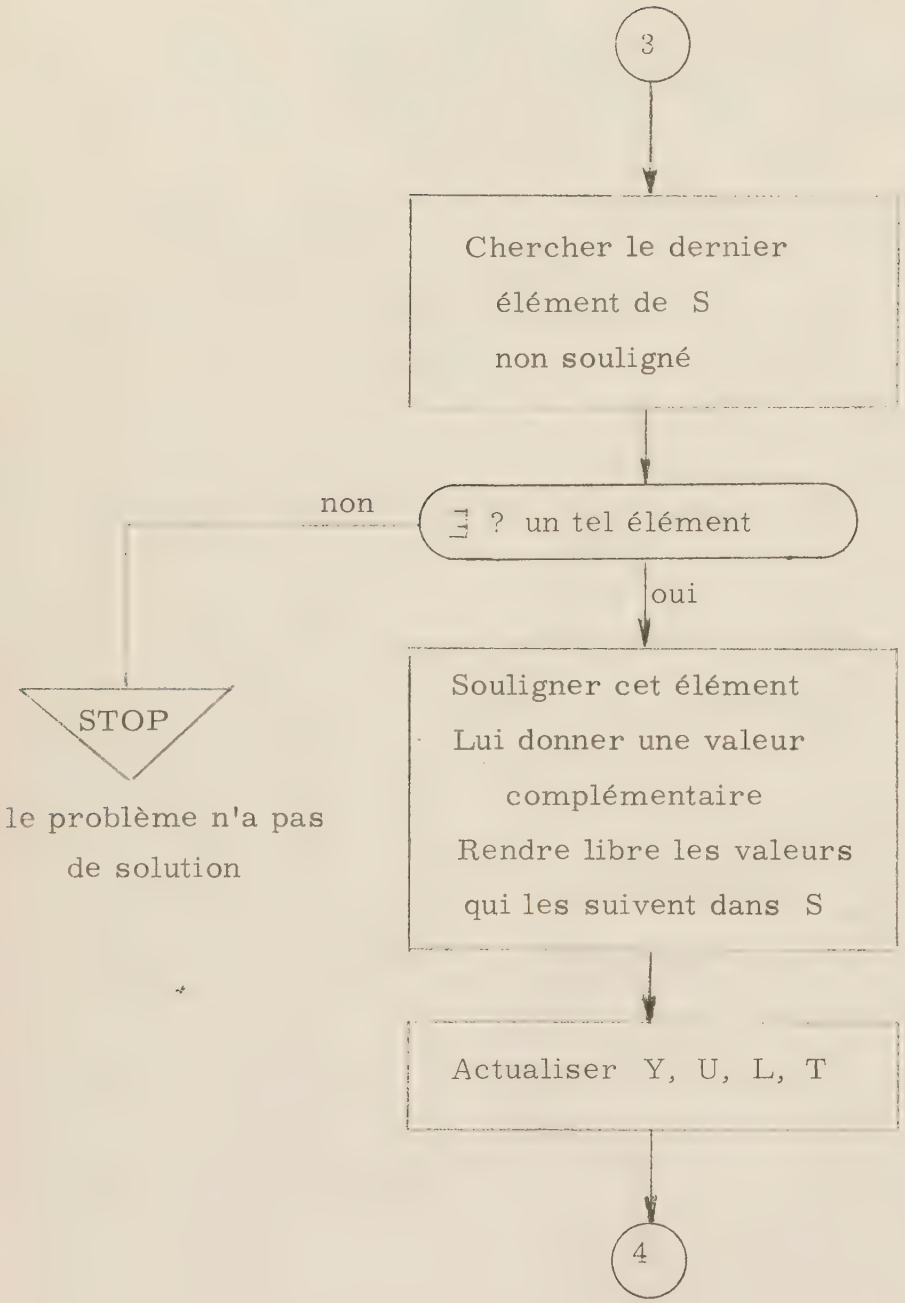
1 - Vérification cyclique



2 - Choix d'une variable dans la solution



3 - Retour en arrière



DEUXIEME PARTIE

EXEMPLES D'APPLICATION

III-2-1 - ETUDE DE RESEAUX CELLULAIRES COMPOSES D'OPERATEURS NI-ON [32, 33]

III-2-1-1 - Définition de la cellule de base du réseau

La cellule de base de la structure cellulaire, représentée sur la figure III-5 est un opérateur NI-ON à deux entrées et à sortie dédoublée.

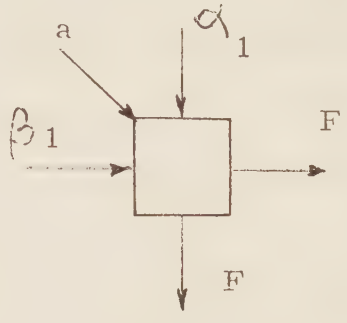


Figure III-5

Cette cellule réalise, en fonction de l'assignement du paramètre "a", soit la fonction logique NI, soit la fonction logique ON de ses deux entrées α_1 et β_1. Nous avons déjà montré dans le chapitre II l'intérêt que présentait un tel type d'opérateur aussi bien dans des synthèses arborescentes que dans les synthèses à l'aide de réseaux en cascade.

L'assignement du paramètre "a" peut se faire, au niveau de la réalisation technologique, par une métallisation supplémentaire qui fixera le potentiel de l'entrée "a", soit au niveau haut, soit au niveau bas.

La fonction logique F réalisée par le module, exprimée en fonction des entrées binaires a, α_1, β_1 est

$$F = \overline{\alpha}_1 . \overline{\beta}_1 + \overline{a} . \overline{\alpha}_1 + \overline{a} . \overline{\beta}_1 \qquad (10)$$

Si a = 0 $\quad F = \overline{\alpha_1} + \overline{\beta_1}$ le module réalise la fonction ON de α_1 et β_1

Si a = 1 $\quad F = \overline{\alpha_1} \cdot \overline{\beta_1}$ le module réalise la fonction NI de α_1 et β_1

Remarquons que la fonction définie par (10) n'est autre que le complément de la fonction majorité de a, α_1 et β_1.

III-2-1-2 - Système d'inégalités associé à la cellule de base

A la fonction logique $F(\alpha_1, \beta_1) = \overline{\alpha_1} \cdot \overline{\beta_1}$ (NI) est associé le système de relations conditionnelles suivant :

$$\text{I} \quad \begin{cases} -\alpha_1^{(j)} - \beta_1^{(j)} + 0 \geqslant 0 & \text{si} \quad F(\alpha_1, \beta_1)^{(j)} = 1 \\ -\alpha_1^{(j)} - \beta_1^{(j)} + 0 \leqslant -1 & \text{si} \quad F(\alpha_1, \beta_1)^{(j)} = 0 \end{cases}$$

A la fonction logique $F(\alpha_1, \beta_1) = \overline{\alpha_1} + \overline{\beta_1}$ (ON) est associé le système de relations conditionnelles suivant :

$$\text{II} \quad \begin{cases} -\alpha_1^{(j)} - \beta_1^{(j)} + 1 \geqslant 0 & \text{si} \quad F(\alpha_1, \beta_1)^{(j)} = 1 \\ -\alpha_1^{(j)} - \beta_1^{(j)} + 0 \leqslant -1 & \text{si} \quad F(\alpha_1, \beta_1)^{(j)} = 0 \end{cases}$$

où $\alpha_1^{(j)}$, $\beta_1^{(j)}$ et $F^{(j)}$ représentent les valeurs respectives des entrées α_1 et β_1 et de la fonction F qui correspondent à une combinaison des entrées dont l'équivalent décimal est j (Table IV)

j	α_1	β_1
0	0	0
1	0	1
2	1	0
3	1	1

Table IV

Compte tenu du fait que la fonction des trois variables a, α_1, β_1 est elle-même linéairement séparable, il est possible de déterminer un système de relations conditionnelles qui regroupent les systèmes I et II.

Ce système est le suivant :

$$\text{III} \quad \begin{cases} -\alpha_1^{(j)} - \beta_1^{(j)} - a + 1 \geqslant 0 & \text{si} \quad F^j(a, \alpha_1, \beta_1) = 1 \\ -\alpha_1^{(j)} - \beta_1^{(j)} - a + 1 \leqslant -1 & \text{si} \quad F^j(a, \alpha_1, \beta_1) = 0 \end{cases}$$

En effet, si a = 0 le système III devient :

$$\text{IIIa} \quad \begin{cases} -\alpha_1^{(j)} - \beta_1^{(j)} + 1 \geqslant 0 & \text{si} \quad F^{(j)}(0, \alpha_1, \beta_1) = 1 \\ -\alpha_1^{(j)} - \beta_1^{(j)} + 1 \leqslant -1 & \text{si} \quad F^{(j)}(0, \alpha_1, \beta_1) = 0 \end{cases}$$

qui est identique au système II.

si a = 1 le système III devient :

$$\text{IIIb} \quad \begin{cases} -\alpha_1^{(j)} - \beta_1^{(j)} + 0 \geqslant 0 & \text{si} \quad F^{(j)}(1, \alpha_1, \beta_1) = 1 \\ -\alpha_1^{(j)} - \beta_1^{(j)} + 0 \leqslant -1 & \text{si} \quad F^j(1, \alpha_1, \beta_1) = 0 \end{cases}$$

qui est identique au système I.

Passage du système III à un système d'inégalités linéaires

En utilisant les règles définies dans la première partie de ce chapitre nous pouvons réécrire le système III comme :

$$\text{IV} \quad \begin{cases} \text{a)} & -\alpha_1^{(j)} - \beta_1^{(j)} - a + 1 \geqslant -U\,(1 - F^{(j)}) \\ \text{b)} & -\alpha_1^{(j)} - \beta_1^{(j)} - a + 1 \leqslant -1 + F^{(j)} \end{cases}$$

La valeur U = 2 convient. En effet :

si $F^{(j)} = 1$ IV a) donne $-\alpha_1^{(j)} - \beta_1^{(j)} - a + 1 \geqslant 0$

IV b) donne $-\alpha_1^{(j)} - \beta_1^{j} - a + 1 \leqslant 1$

qui est forcément vérifiée

Si $F^{(j)} = 0$ IV a) donne $-\alpha_1^{(j)} - \beta_1^{(j)} - a + 1 \geqslant -2$

qui est forcément vérifiée

IV b) donne $-\alpha_1^{(j)} - \beta_1^{(j)} - a + 1 \leqslant -1$

donc le système d'inégalités qui décrit le fonctionnement du module de base est :

$$\text{V} \quad \begin{cases} -\alpha_1^{(j)} - \beta_1^{(j)} - a + 1 \geqslant -2\,(1 - F^{(j)}) \\ -\alpha_1^{(j)} + \beta_1^{(j)} + a - 1 \geqslant 1 - 2\,F^{(j)} \end{cases}$$

III-2-1-3 - Application à une structure régulière de 4 cellules de base

La structure étudiée est représentée sur la figure III-6

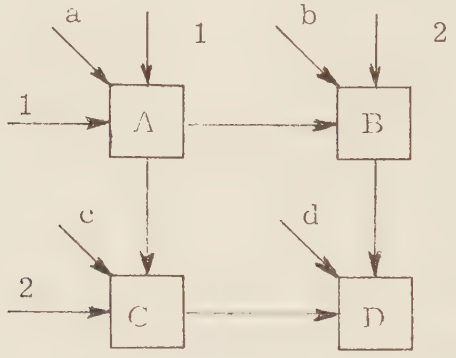


Figure III-6

Soient a, b, c, d les assignements des modules A, B, C et D et soient $A^{(j)}$, $B^{(j)}$, $C^{(j)}$ et $D^{(j)}$ les sorties de ces modules. Les résultats des paragraphes précédents permettent de décrire cette structure par le système d'inégalités :

Module A

$$\begin{cases} -\alpha_1^{(j)} - \beta_1^{(j)} - a + 1 \geqslant -2\,(1 - A^{(j)}) \\ \alpha_1^{(j)} + \beta_1^{(j)} + a - 1 \geqslant 1 - 2\,A^{(j)} \end{cases}$$

Module B

$$\begin{cases} -\alpha_2^{(j)} - A^{(j)} - b + 1 \geqslant -2\,(1 - B^{(j)}) \\ \alpha_2^{(j)} + A^{(j)} + b - 1 \geqslant 1 - 2\,B^{(j)} \end{cases}$$

Module C

$$\begin{cases} -\beta_2^{(j)} - A^{(j)} - c + 1 \geqslant -2\,(1 - C^{(j)}) \\ \beta_2^{(j)} + A^{j} + c - 1 \geqslant 1 - 2\,C^{(j)} \end{cases}$$

Module D $\left[\begin{array}{l} -B^{(j)} - C^{(j)} - d + 1 \geqslant -2\,(1 - D^{(j)}) \\ \\ B^{(j)} + C^{(j)} + d - 1 \geqslant 1 - 2\,D^{(j)} \end{array}\right.$

Dans la plupart des cas la fonction logique dont nous voulons faire la synthèse sera réalisée en sortie de la cellule de base la plus à droite et la plus en bas du réseau cellulaire étudié, c'est-à-dire D pour le réseau de la figure III-5. Les valeurs de la sortie $D^{(j)}$ pour les différentes combinaisons possibles des variables d'entrée seront donc des données dans la résolution du système d'inégalités associé au réseau cellulaire. Les inconnues du problème peuvent se ranger en trois catégories :

a - les entrées d'assignement a, b, c, d de chacune des cellules du réseau

b - les valeurs des sorties intermédiaires $A^{(j)}$, $B^{(j)}$ et $C^{(j)}$ pour les différentes combinaisons des entrées

c - les assignements des entrées α_1, α_2, β_1 et β_2 qui peuvent se faire par un choix dans différents ensembles de variables d'entrée. Les dimensions de ces ensembles sont bien sur déterminées par le nombre de variables de la fonction à réaliser mais aussi par le fait que nous restons libre de déterminer sous quelle forme ces variables booléennes attaqueront les entrées du réseau : soit uniquement sous leur forme vraie soit aussi sous leur forme complémentée. De plus nous pouvons introduire la possibilité d'utiliser comme entrées de réseau les constantes 0 ou 1. Ces problèmes sont d'ailleurs très liés au problèmes de réalisations technologiques. Du choix que nous aurons fait dépendront les dimensions plus ou moins importantes du système que nous devrons résoudre mais aussi des possibilites logiques plus ou moins grandes pour le réseau considéré.

Supposons que nous voulions réaliser avec le réseau de la figure des fonctions de deux variables x_1 et x_2 et que ses deux variables puissent se présenter aux entrées du réseau sous leur forme vraie ou complémentée. Nous aurons donc à choisir chacune des entrées α_1, α_2, β_1 et β_2 dans l'ensemble X $\{x_1, x_2, \overline{x}_1, \overline{x}_2\}$. Nous devons donc traduire algébriquement cette propriété des entrées du réseau de pouvoir être affectées à une variable d'un ensemble. Pour cela numérotons 1 à i les entrées externes du réseau et 1 à k les éléments de l'ensemble X. Définissons λ_i^k de la façon suivante :

Si l'entrée i du réseau est connectée à l'élément k de l'ensemble X $\lambda_i^k = 1$

Si non $\lambda_i^k = 0$

Nous pouvons alors remplacer dans le système associé au réseau chaque entrée par une combinaison linéaire de la forme :

$$_i^{(j)} = \lambda_i^1 x_1^{(j)} + \lambda_i^2 x_2^j + \lambda_i^3 (1 - x_1^{(j)}) + \lambda_i^4 (1 - x_2^{(j)})$$

les λ_i^k vérifiant les relations :

$$\sum_k \lambda_i^k = 1$$

En effet chacune des entrées ne peut recevoir qu'une seule variable ou constante booléenne.

III-2-1-4 - Dimensions du problème

La démarche précédente peut être étendue à des structures régulières de dimensions quelconques

Soit m le nombre de cellules de base
Soit l le nombre d'entrées externes
Soit n le nombre de variables d'entrée

La matrice M du système associé sera alors de dimensions p x q avec :

$$p = m + (m - 1)\ 2^n + l^2$$

$$q = 2\ m\ 2^n + 2\ l$$

Les dimensions du système augmentent donc relativement vite avec les dimensions du réseau. Nous donnons ci-dessous les résultats obtenus avec un programme utilisant le deuxième algorithme décrit dans le première partie de ce chapitre.

III-2-1-5 - Résultats expérimentaux

III-2-1-5-1 - Réseau de 4 cellules

Ce réseau est représenté sur la figure III-7.

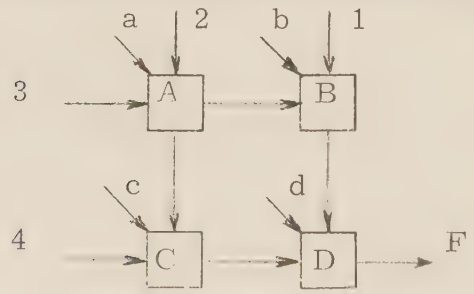


Figure III-7

Il possède quatre entrées externes 1, 2, 3 et 4 et quatre entrées de spécification a, b, c, d. Si nous supposons que les entrées externes possibles sont choisies dans l'ensemble $\{x_1, \overline{x}_1, x_2, \overline{x}_2\}$ les dimensions de la matrice M du système associé d'inégalités seront p = 32 et q = 40. Le programme détermine pour différentes fonctions F les variables auxquelles sont assignées les entrées externes ainsi que les valeurs des entrées de spécification a, b, c, d. Le tableau IV regroupe les résultats obtenus pour les seize fonctions de 2 variables.

III-2-1-5-2 - Réseau de neuf cellules

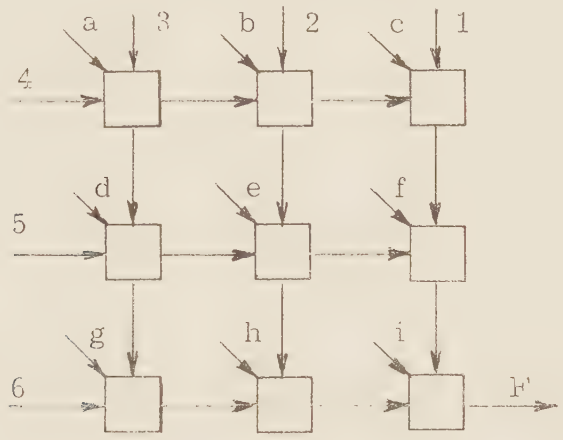


Figure III-8

Il possède six entrées externes et neuf entrées de spécification. Si les entrées externes sont choisies dans l'ensemble $\{x_1, \overline{x}_1, x_2, \overline{x}_2, x_3, \overline{x}_3\}$ les dimensions de la matrice M du système associé d'inégalités sont : p = 109 et q = 156. Le tableau V donne des indications sur le nombre d'itérations nécessaires pour réaliser certaines fonctions de trois variables. Nous avons omis pour simplifier le tableau les affichages des entrées externes et des entrées de spéficiations qui sont aussi calculées par le programme.

Fonctions à réaliser	Entrées externes				Entrées de spécification				Nombre d'itérations
	1	2	3	4	a	b	c	d	
$F_1 = 0$	x_1	x_1	x_1	x_1	0	1	0	1	4
$F_2 = x_1 \cdot x_2$	x_1	$\overline{x}_1$	$\overline{x}_1$	x_2	0	1	0	1	6
$F_3 = x_1 \cdot \overline{x}_2$	x_1	x_2	x_1	x_2	0	1	0	1	6
$F_4 = x_1$	$\overline{x}_1$	$\overline{x}_1$	$\overline{x}_1$	x_1	0	0	0	0	5
$F_5 = \overline{x}_1 x_2$	x_1	x_1	x_1	x_2	0	1	0	1	6
$F_6 = x_2$	$\overline{x}_2$	$\overline{x}_2$	$\overline{x}_2$	$\overline{x}_2$	0	0	0	0	5
$F_7 = \overline{x}_1 x_2 + x_1 \overline{x}_2$	$\overline{x}_2$	$\overline{x}_2$	x_1	x_1	1	1	1	1	23
$F_8 = x_1 + x_2$	$\overline{x}_1$	$\overline{x}_1$	$\overline{x}_2$	x_1	0	1	1	1	6
$F_9 = \overline{x}_1 \cdot \overline{x}_2$	x_1	x_1	x_1	$\overline{x}_2$	0	1	0	1	6
$F_{10} = \overline{x}_1 \overline{x}_2 + x_1 x_2$	x_2	x_2	x_1	x_1	1	1	1	1	20
$F_{11} = \overline{x}_2$	x_2	x_2	x_2	$\overline{x}_2$	0	0	0	0	5
$F_{12} = x_1 + \overline{x}_2$	x_2	x_2	x_2	x_1	0	1	1	1	6
$F_{13} = \overline{x}_1$	x_1	x_1	x_1	$\overline{x}_1$	0	0	0	0	5
$F_{14} = \overline{x}_1 + x_2$	x_1	x_1	x_1	x_2	0	1	1	1	14
$F_{15} = \overline{x}_1 + \overline{x}_2$	x_1	x_1	x_2	$\overline{x}_1$	0	1	1	1	22
$F_{16} = 1$	x_1	x_1	x_1	x_1	0	1	1	1	5

Tableau IV

Fonctions à réaliser	Nombre d'itérations
$F = x_2 + \bar{x}_1 x_3$	186
$F = \bar{x}_1 \bar{x}_2 \bar{x}_3$	50
$F = \bar{x}_1 (\bar{x}_2 + x_3)$	44
$F = \bar{x}_1 \bar{x}_2 + x_1 x_2 x_3$	1563
$F = \bar{x}_2 + \bar{x}_1 \bar{x}_3$	17
$F = x_3 (\bar{x}_2 + x_1)$	3099
$F = \bar{x}_2 (\bar{x}_1 + x_3)$	59

Tableau V

III-2-1-5-3 - Conclusions

Nous pourrions multiplier le nombre d'exemples mais notre but est surtout de montrer la faisabilité des méthodes et algorithmes mis au point. Nous avons choisi volontairement le nombre d'itérations comme critère de jugement de l'efficacité de l'algorithme. C'est un critère général dans la mesure où il ne dépend pas du type du calculateur utilisé. Nous constatons que ce nombre varie dans de grandes proportions en fonction de l'exemple traité ce qui limite les performances de la méthode. Une autre contrainte concerne les dimensions de la matrice associée M ce qui justifie la mise en place de procédures de simplification.

III-2-1-6 - Réduction des dimensions du problème : Simplification du réseau

Les dimensions du système associé au réseau peuvent être réduites en tenant compte des propriétés des opérateurs majorité complémentée placés en cascade qui se traduisent par le théorème suivant :

III-2-1-6-1 - Théorème : n opérateurs majorité complémentée placés en série sont équivalents à un seul élément à seuil.

Démonstration

Cette démonstration est basée sur les différents théorèmes que nous avons énoncés au début de ce chapitre. Nous raisonnerons par induction

a - un seul module majorité complémentée (figure III-9)

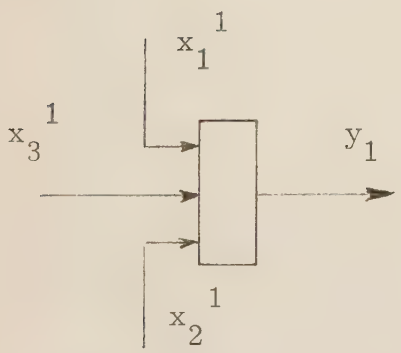


Figure III-9

$$y_1 = \bar{x}_1^1 \cdot \bar{x}_2^1 + (\bar{x}_1^1 + \bar{x}_2^1) \cdot \bar{x}_3^1$$

$$= \bar{x}_1^1 \cdot \bar{x}_2^1 + (\bar{x}_1^1 + \bar{x}_2^1) \cdot y_0$$

$$y_0 = \bar{x}_3^1 = \langle -x_3^1 \rangle_{0:-1} = \langle \bar{x}_3^1 \rangle_{1:0}$$

$$u_0 = 1 \quad l_0 = 0 \quad \sigma_{p0} = 1 \quad \sigma_{n0} = 0 \quad a_0 = 1$$

Donc d'après le théorème III :

$$y_1 = \langle \bar{x}_1^1 + \bar{x}_2^1 + \bar{x}_3^1 \rangle_{2:1}$$

d'après le théorème V :

$$y_1 = \langle 3 - x_1^{\,1} - x_2^{\,1} - x_3^{\,1} \rangle_{2:1} \qquad b_0 = 3$$

$$= \langle - x_1^{\,1} - x_2^{\,1} - x_3^{\,1} \rangle_{-1:-2}$$

donc : $U_1 = -1 = u_0 + a_0 - b_0$

$L_1 = -2 = l_0 + a_0 - b_0$

b - deux modules en série

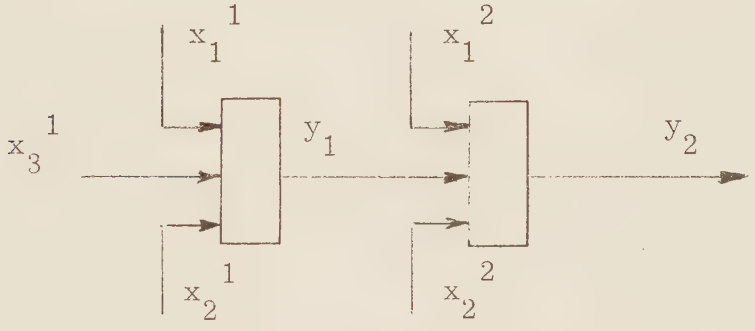


Figure III-10

$$y_2 = \overline{x_1}^{\,2}\,\overline{x_2}^{\,2} + (\overline{x_1}^{\,2} + \overline{x_2}^{\,2})\,\overline{y_1}$$

$$\overline{y_1} = \langle x_1^{\,1} + x_2^{\,1} + x_3^{\,1} \rangle_{2:1}$$

$u_1 = 2 \quad l_1 = 1 \quad \sigma_{p1} = 3 \quad \sigma_{n1} = 0 \quad a_1 = 2$

$$y_2 = \langle 2^1 \bar{x}_1^2 + 2^1 \bar{x}_2^2 + x_1^1 + x_2^1 + x_3^1 \rangle_{4:3}$$

$$= \langle 4 - 2^1 x_1^2 - 2^1 x_2^2 + x_1^1 + x_2^1 + x_3^1 \rangle_{4:3} \qquad b_1 = 4$$

$$= \langle - 2^1 x_1^2 - 2^1 x_2^2 + x_1^1 + x_2^1 + x_3^1 \rangle_{0:-1}$$

donc :

$$U_2 = u_1 + a_1 - b_1 \qquad L_2 = l_1 + a_1 - b_1$$

avec $u_1 = - L_1$ et $l_1 = - U_1$

c - trois modules en série

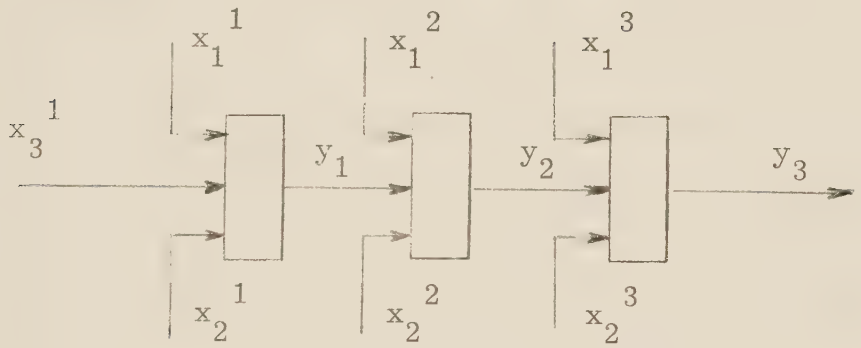


Figure III-11

$$y_3 = \bar{x}_1^3 \cdot \bar{x}_2^3 + (\bar{x}_1^3 + \bar{x}_2^3)\, \bar{y}_2$$

$$\bar{y}_2 = \langle 2^1 x_1^2 + 2^1 x_2^2 - x_1^1 - x_2^1 - x_3^1 \rangle_{1:0}$$

$u_2 = 1$ $l_2 = 0$ $\sigma_{p2} = 4$ $\sigma_{n2} = - 3$ $a_2 = 4$

$$y_3 = \langle 2^2 \bar{x}_1^{\,3} + 2^2 \bar{x}_2^{\,3} + 2^1 x_1^{\,2} + 2^1 x_2^{\,2} - x_1^{\,1} - x_2^{\,1} - x_3^{\,1} \rangle_{5:4}$$

$$= \langle 8 - 2^2 x_1^{\,3} - 2^2 x_2^{\,3} + 2^1 x_1^{\,2} + 2^1 x_2^{\,2} - x_1^{\,1} - x_2^{\,1} - x_3^{\,1} \rangle_{5:4} \qquad b_2 = 8$$

$$= \langle -2^2 x_1^{\,3} - 2^2 x_2^{\,3} + 2^1 x_1^{\,2} + 2^1 x_2^{\,2} - x_1^{\,1} - x_2^{\,1} - x_3^{\,1} \rangle_{-3:4}$$

$$U_3 = u_2 + a_2 - b_2 \qquad\qquad L_3 = l_2 + a_2 - b_2$$

$$u_2 = -L_2 \qquad\qquad l_2 = -U_2$$

d - Supposons les résultats précédents vérifiés pour n modules en série :

alors

$$y_n = \langle (-1)^{n+1} \sum_{i=1}^{n} (-1)^i 2^{i-1} (x_1^{\,i} + x_2^{\,i}) + (-1)^n x_3^{\,1} \rangle_{U_n : L_n}$$

avec

$$U_n = \frac{4 - 2^n}{3} \qquad \text{pour n pair} \qquad L_n = U_n - 1$$

$$U_n = \frac{-1 - 2^n}{3} \qquad \text{pour n impair} \qquad L_n = U_n - 1$$

e - Il est alors aisé de montrer que ces résultats sont vérifiés dans le cas de n + 1 modules en série c'est-à-dire que nous retrouvons pour y_{n+1} une expressions identique à celle de y_n.

En définitive n éléments majorité complémentée à trois entrées placées en cascade (figure III-12) sont équivalents à un seul élément à seuil (figure III-13) dont les poids et le seuil sont définis de la façon suivante :

Soit i l'indice affecté au nombre de modules : i varie de 1 à n

Soit j l'indice affecté à chaque module de la cascade : j varie de 1 à i

Soit w_i^j le poids affecté aux entrées de l'élément à seuil équivalent et qui correspondent aux entrées x^j de l'élément j de la cascade.

Soit T le seuil de l'élément équivalent

On a :

$$T = \frac{4 - 2^i}{3} \qquad \text{si i est pair}$$

$$T = \frac{-1 - 2^i}{3} \qquad \text{si i est impair}$$

et

$$w_i^j = (-1)^{i+j-1} \; 2^{j-1}$$

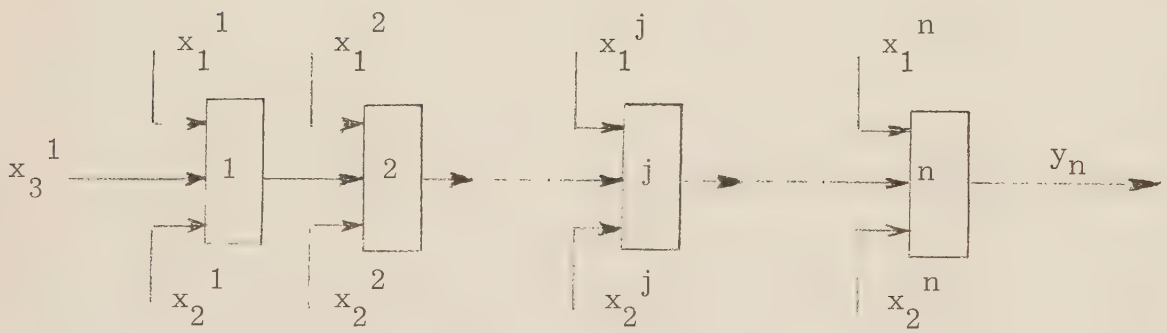


Figure III-12

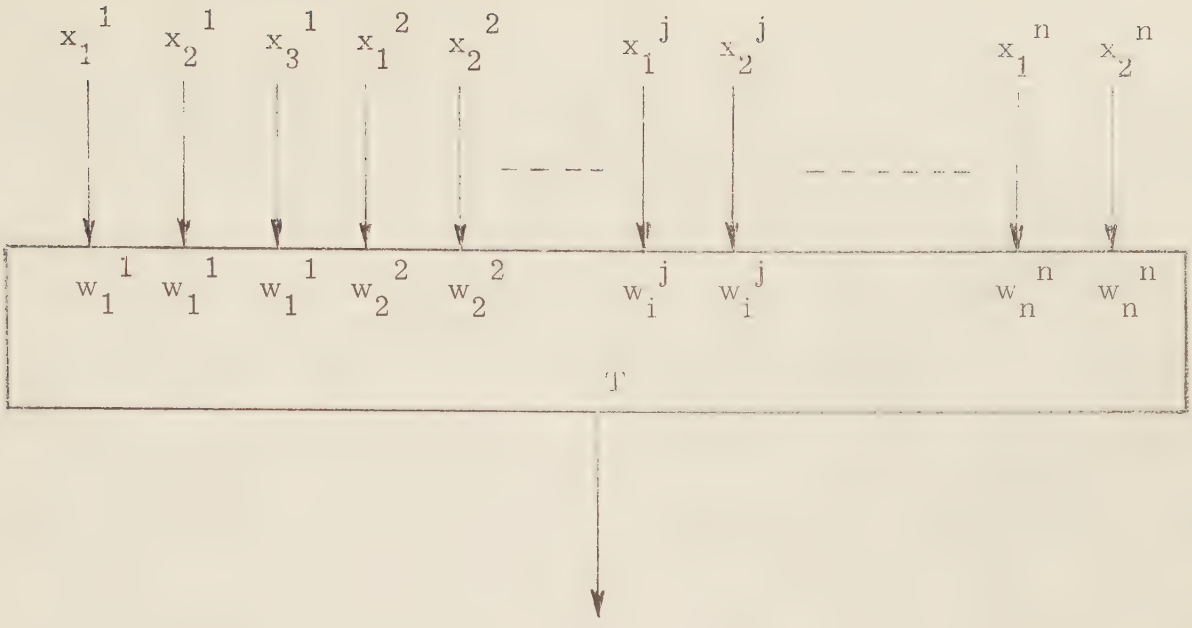


Figure III-13

III-2-1-6-2 - Conséquences du théorème précédent

La propriété précédente permet de représenter une structure régulière par un réseau d'élément s à seuil dont le nombre d'éléments m_s sera inférieur au nombre d'éléments m de la structure. Le rapport m/m_s est voisin de 2. Le tableau V donne les correspondances exactes entre m et m_s.

Nombre de modules de la structure : m	4	9	16	25	36
Nombre d'éléments à seuil : m_s	2	4	7	11	16

Tableau V

Nous détaillons sur la figure III-14 le passage de la structure régulière au réseau d'éléments à seuil pour le cas où m = 4

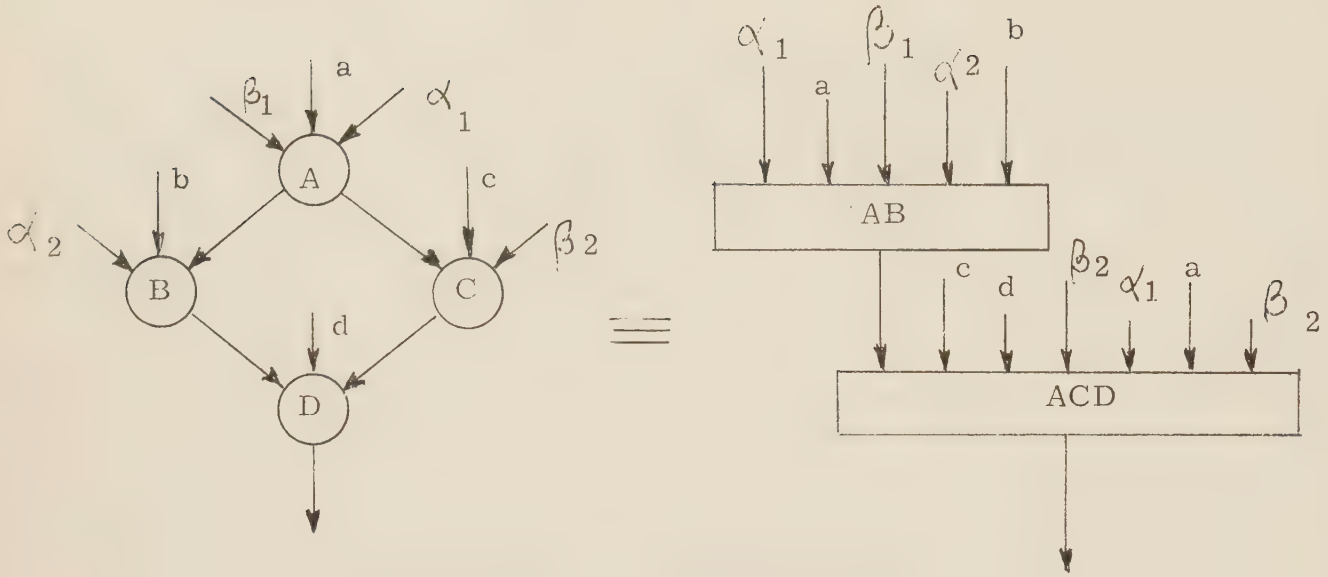


Figure III-14

Le système associé au réseau est alors :

AB

$$\begin{cases} +\alpha_1^j + a + \beta_1^j - 2\alpha_2^j - 2b \geqslant -4(1-B^j) \\ -\alpha_1^j - a - \beta_1^j + 2\alpha_2^j + 2b \geqslant 1 - 4B^j \end{cases}$$

ACD

$$\begin{cases} -\alpha_1^j - a - \beta_1^j + 2\beta_2^j + 2c - 4d - 4B^j - 3 \geqslant -8(1-D^j) \\ \alpha_1^j + a + \beta_1^j - 2\beta_2^j - 2c + 4d + 4B^j + 3 \geqslant 1 - 8B^j \end{cases}$$

Nous donnons dans le tableau VI les dimensions comparatives de la matrice A du système associé dans le cas d'une structure régulière avant et après simplification.

Nombre de cellules de la structure régulière	Dimensions de A système non simplifié	Dimensions de A système simplifié
4	32 x 40	24 x 24
9	109 x 156	69 x 76
16	320 x 528	176 x 240
25	893 x 820	445 x 372

Tableau VI

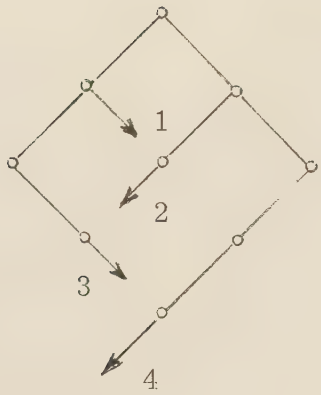
La figure III-15 indique les équivalences entre chaque réseau pour $m > 4$: les connexions ont été interrompues pour marquer que les éléments placés en amont sont regroupés pour constituer un seul élément à seuil.

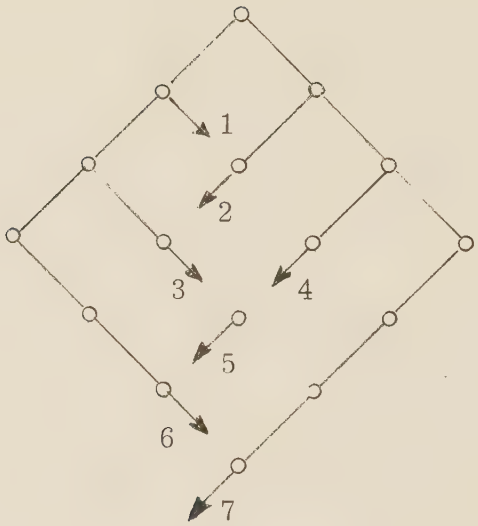
III-2-1-6-3 - Résultats expérimentaux

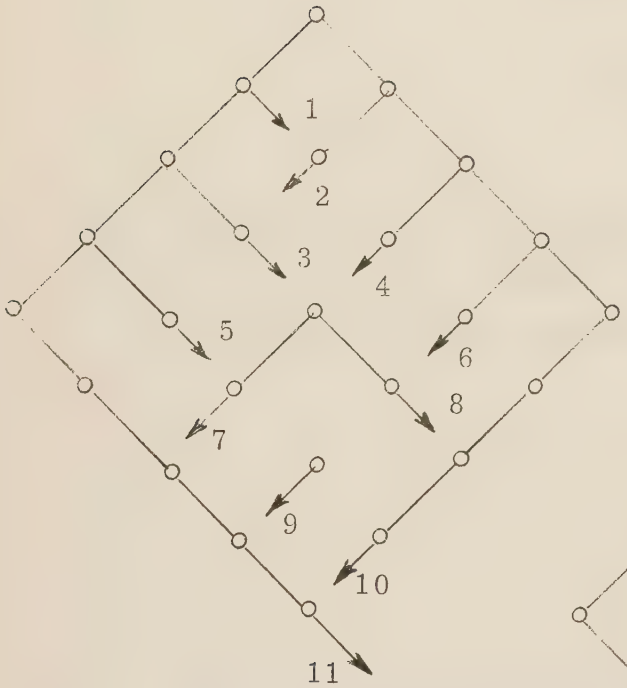
A - Réseau de quatre cellules

Le réseau est le même que celui de la figure III-16. Le tableau VI regroupe les résultats obtenus par le programme pour les fonctions de deux variables.

B - Réseau de neuf cellules (Tableau VII)







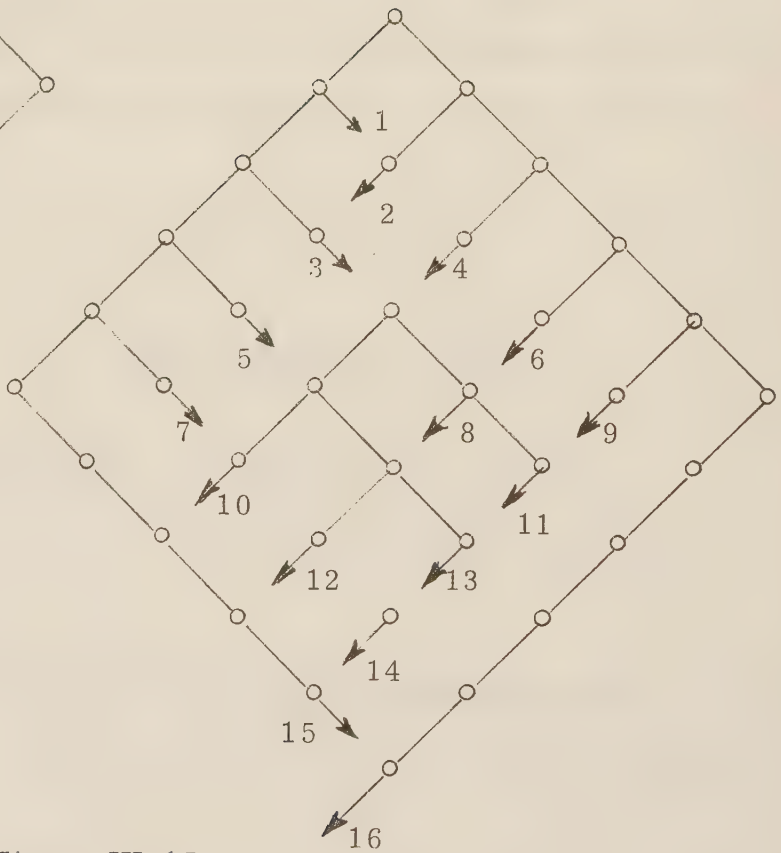


Figure III-15

Fonctions à réaliser	Entrées externes 1	2	3	4	Entrées de spécification a	b	c	d	Nombre d'itérations
$F_1=0$	x_1	x_1	$\overline{x}_1$	$\overline{x}_1$	1	1	1	0	8
$F_2=x_1.x_2$	x_1	$\overline{x}_1$	$\overline{x}_2$	x_2	1	1	1	0	6
$F_3=x_1.\overline{x}_2$	x_1	x_2	x_2	x_1	0	1	1	0	10
$f_4=x_1$	x_1	$\overline{x}_1$	$\overline{x}_1$	x_1	0	1	1	0	10
$F_5=\overline{x}_1.x_2$	x_1	x_1	x_1	x_2	0	1	1	1	6
$F_6=x_2$	x_1	$\overline{x}_2$	$\overline{x}_2$	x_2	0	1	0	1	7
$F_7=\overline{x}_1.x_2+x_1\overline{x}_2$	x_1	x_2	x_1	x_2	1	1	1	1	15
$F_8=x_1+x_2$	x_1	$\overline{x}_1$	$\overline{x}_2$	x_1	1	1	0	1	7
$F_9=\overline{x}_1.\overline{x}_2$	x_1	x_1	x_1	$\overline{x}_2$	0	1	1	1	6
$F_{10}=\overline{x}_1\overline{x}_2+x_1x_2$	x_1	$\overline{x}_2$	x_1	$\overline{x}_2$	1	1	1	1	21
$F_{11}=\overline{x}_2$	x_1	x_2	x_2	$\overline{x}_2$	0	1	0	1	7
$F_{12}=x_1+\overline{x}_2$	x_1	x_2	$\overline{x}_1$	x_1	1	1	0	1	7
$F_{13}=\overline{x}_1$	x_1	x_1	x_1	$\overline{x}_1$	0	1	1	1	6
$F_{14}=\overline{x}_1+x_2$	x_1	x_1	x_2	$\overline{x}_1$	1	1	1	1	8
$F_{15}=\overline{x}_1+\overline{x}_2$	x_1	x_1	x_2	$\overline{x}_1$	1	1	1	1	8
$F_{16}=1$	x_1	x_1	$\overline{x}_1$	$\overline{x}_1$	1	1	1	1	10

Tableau VI

Fonctions à réaliser	Nombre d'itérations
$F = \bar{x}_1 \bar{x}_2 \bar{x}_3$	110
$F = \bar{x}_1 (\bar{x}_2 + x_3)$	199
$F = \bar{x}_2 + \bar{x}_1 x_3$	154
$F = x_1 x_2 + x_1 x_3 + x_2 x_3$	192
$F = x_3 + x_1 x_2$	19
$F = x_1 + x_2 \bar{x}_3$	78
$F = x_2 (x_3 + x_1)$	194

Dimensions de la matrice M associée : 69 x 76

Tableau VII

C - Conclusions

La simplification du réseau par mise en cascade du réseau se traduit surtout par la diminution des capacités mémoires nécessaires pour traiter le problème, ce qui permet de traiter des fonctions et des réseaux plus complexes.

III-2-2 - ETUDE DE RESEAUX CELLULAIRES COMPOSES D'OPERATEURS MAJORITE

III-2-2-1 - Description de la cellule de base

Le cellule de base a la même structure externe que celle de la figure III-5 c'est-à-dire qu'elle possède trois entrées α_1, β_1 et γ_1 et une sortie dédoublée. Les trois entrées pourront être assignées dans l'ensemble des variables de la fonction dont nous faisons la synthèse.

La fonction logique réalisée par la cellule de base est la fonction majorité de trois variables dont l'expression est :

$$F = \alpha_1 \beta_1 + \alpha_1 \gamma_1 + \beta_1 \gamma_1 \qquad (11)$$

La sortie de l'opérateur vaut 1 lorsque deux ou trois de ses entrées valent 1. De plus la fonction décrite par (11) est totalement symétrique.

III-2-2-2- Système d'inégalités associé à la cellule de base

A la fonction logique $F = \alpha_1 \beta_1 + \beta_1 \gamma_1 + \alpha_1 \gamma_1$ est associée le système de relations conditionnelles :

$$I \quad \begin{cases} \alpha_1^{(j)} + \beta_1^{(j)} + \gamma_1^{(j)} - 2 \geqslant 0 & \text{si} \quad F^{(j)} = 1 \\ \alpha_1^{(j)} + \beta_1^{(j)} + \gamma_1^{(j)} - 2 \leqslant -1 & \text{si} \quad F^{(j)} = 0 \end{cases}$$

En utilisant les règles définies précédemment ce système de relations conditionnelles se transforme dans le système d'inégalités :

$$\text{II} \quad \begin{cases} \alpha_1^{(j)} + \beta_1^{(j)} + \gamma_1^{(j)} - 2 \geqslant -2\,(1 - F^{(j)}) \\ \alpha_1^{(j)} + \beta_1^{(j)} + \gamma_1^{(j)} - 2 \leqslant -1 + 2\,F^{(j)} \end{cases}$$

III-2-2-3 - Résultats expérimentaux

Connaissant le système d'inégalités associé à la cellule de base nous pouvons décrire une structure régulière de dimensions quelconques. Notons que la propriété décrite précédemment de possibilité de remplacement d'une cascade par un seul élément à seuil est aussi valable pour les opérateurs majorité.

Pour le réseau de quatre cellules majorités de la figure III-16 par exemple les résultats obtenus pour des fonctions de trois variables sont consignées dans le tableau VII.

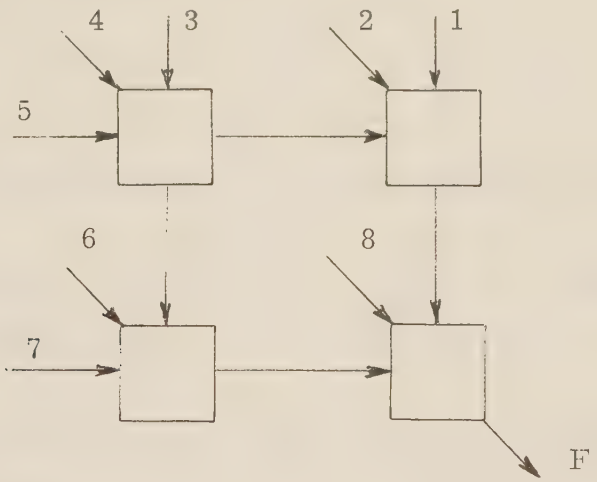


Figure III-16

Fonctions à réaliser	Affichage des entrées								Nombre d'itérations
	1	2	3	4	5	6	7	8	
$F=x_1\bar{x}_2\bar{x}_3+x_2x_3$	x_3	$\bar{x}_3$	x_2	1	$\bar{x}_2$	0	x_1	x_1	308
$F=x_1x_2+x_3$	x_2	1	1	x_3	$\bar{x}_2$	x_3	x_1	x_1	44
$F=x_1x_2+\bar{x}_1x_3$	x_2	1	1	x_2	x_1	x_3	x_1	x_1	67
$F=\bar{x}_1x_2x_3+x_1\bar{x}_2+\bar{x}_3$	$\bar{x}_2$	1	x_2	x_2	x_1	x_3	x_1	x_1	190
$F=\bar{x}_1\bar{x}_2\bar{x}_3+\bar{x}_1x_2x_3$ $+x_1\bar{x}_2x_3+x_1x_2\bar{x}_3$	x_1	x_3	x_2	1	x_2	$\bar{x}_2$	x_1	x_1	305
$F=x_1+x_2x_3$	1	1	x_1	x_3	$\bar{x}_3$	x_3	x_1	x_1	16
$F=x_1x_2x_3$	x_2	1	x_3	0	x_1	x_1	x_1	x_1	104
$F=x_1x_2x_3+\bar{x}_1\bar{x}_2\bar{x}_3$	x_2	1	x_2	0	x_1	x_1	x_1	x_1	89

Dimensions de la matrice M associée 80 x 88 : entrées 0 et 1 permises

Tableau VII

III-2-2-4 - Propriétés particulières des réseaux d'opérateurs majorité [34]

Définition I : Un élément à m entrées et une sortie est autodual si la fonction de m variables produite à sa sortie et autoduale

L'élément majorité à trois entrées est un élément autodual.

Définition II : Nous dirons qu'un réseau à n entrées et une sortie est autodual si la fonction de n variables produite à sa sortie est autoduale

Propriété I : Un réseau composé d'éléments autoduaux est autodual. Un réseau composé d'opérateurs majorité est donc un réseau autodual

Propriété II : Toute fonction autoduale de n variables reste autoduale quand deux ou plusieurs de ses arguments sont rendus identiques.

Remarque :

En dépit de la propriété I un réseau d'opérateurs majorité peut générer des fonctions non autoduales. Ceci est dû au fait que si dans une fonction autoduale de n variables nous remplaçons une des variables par une constante, la fonction résultante de (n-1) variables ne sera pas en général autoduale.

Propriété III : Si chaque entrée d'un réseau autodual est une variable, alors sa fonction de sortie sera autoduale

Définition III : Soit F une fonction booléenne des n variables $x_1, \ldots, x_n$ et soit U une variable indépendante de $x_1, \ldots, x_n$, l'expression autoduale F^{ad} pour F est définie comme une fonction de n + 1 variables

$$F^{ad} = U F + U F^{d}$$

Propriété IV : Soit $G(x_1, \ldots, x_n)$ une fonction autoduale de m variables, et soit $F(x_1, \ldots, x_n)$ une fonction de n variables, $n < m$, obtenue à partir de G en rendant les variables $x_{n+1}, \ldots, x_m$ égales aux constantes $C_1, \ldots, C_{m-n}$:

$$F(x_1, \ldots, x_n) = G(x_1, \ldots, x_n, C_1, \ldots, C_{m-n})$$

Alors

$$F^{d}(x_1, \ldots, x_n) = G(x_1, \ldots, x_n, \overline{C}_1, \ldots, \overline{C}_{m-n})$$

Conséquences de la propriété IV :

Propriété V : Si dans un réseau autodual éventuellement avec des entrées constantes produisant une fonction $F(x_1, \ldots, x_n)$, chaque entrée constante est complémentée, le réseau correspondant réalise F^{d}.

Propriété VI : Si un réseau autodual réalise toute fonction autoduale de (n+1) variables il réalise aussi toute fonction de n variables. Inversement un réseau autodual qui réalise toute fonction de n variables réalise aussi toute fonction autoduale de (n+1) variables.

Propriété VII : Un réseau autodual qui réalise une fonction donnée $F(x_1, \ldots, x_n)$ réalise aussi l'expression autoduale $F^{ad}(U, x_1, \ldots, x_n)$ de cette fonction.

Toutes ces propriétés qui s'appliquent directement aux réseaux d'opérateurs majorité permettent de simplifier l'étude de ces réseaux.

III-2-3 - RESEAUX CELLULAIRES COMPOSES D'OPERATEURS PARTICULIERS

III-2-3-1 - Définition de la cellule de base

La cellule de base est représentée sur la figure III-17.

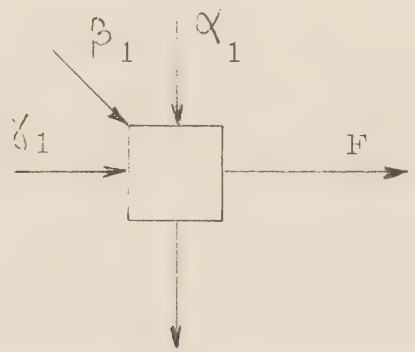


Figure III-17

Sa sortie réalise la fonction booléenne des trois variables α_1, β_1, γ_1 dont l'expression logique est :

$$F = \overline{\alpha}_1 + \overline{\beta}_1 \cdot \overline{\gamma}_1 \qquad (12)$$

C'est un opérateur complet et totalement asymétrique (Annexe I)

III-2-3-2 - Description du réseau par un système d'inégalités

Le système d'inégalités associé à la cellule de base peut se mettre sous la forme :

$$\begin{cases} 5 - 2\alpha_1^{(j)} - \beta_1^{(j)} - \gamma_1^{(j)} - 3F^{(j)} \geqslant 0 \\ -3 + 2\alpha_1^{(j)} + \beta_1^{(j)} + \gamma_1^{(j)} + 3F^{(j)} \geqslant 0 \end{cases}$$

ce qui nous permet de décrire un réseau quelconque composé d'opérateurs de ce type.

III-2-3-3 - Résultats expérimentaux

Les performances de l'algorithme pour ce type de cellules de base ont été testées sur le réseau de la figure III-18.

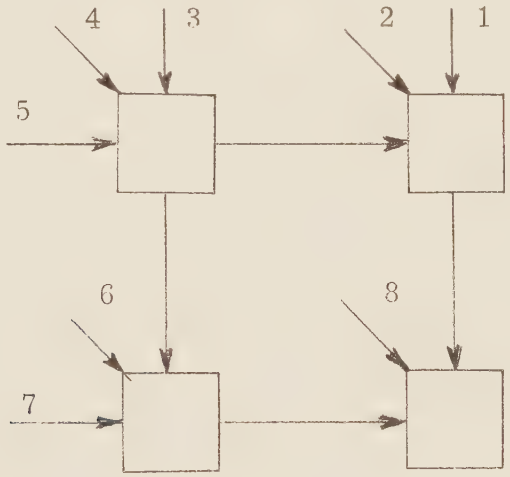


Figure III-18

Les résultats obtenus pour des fonctions de trois variables sont consignés dans le tableau VIII.

Fonctions à réaliser	Affichage des entrées externes								Nombre d'itérations
$F_1 = x_1 x_2 x_3$	x_1	$\overline{x}_1$	x_1	$\overline{x}_1$	x_1	$\overline{x}_2$	x_3	$\overline{x}_1$	109
$F_2 = x_1 x_2 x_3 + \overline{x}_1 \overline{x}_2 \overline{x}_3$	x_1	$\overline{x}_1$	x_3	$\overline{x}_3$	$\overline{x}_2$	x_1	$\overline{x}_3$	x_2	412
$F_3 = x_1 x_2 x_3 + \overline{x}_2 \overline{x}_3$	x_1	$\overline{x}_1$	x_2	x_3	x_1	x_2	$\overline{x}_3$	$\overline{x}_1$	478
$F_4 = \overline{x}_1 x_2 x_3 + x_1 \overline{x}_2 x_3 + x_1 x_2 \overline{x}_3$	x_1	$\overline{x}_1$	x_3	$\overline{x}_2$	x_1	x_3	x_1	$\overline{x}_2$	204
$F_5 = x_1 \overline{x}_2 \overline{x}_3 + x_2 x_3$	x_1	$\overline{x}_1$	$\overline{x}_2$	$\overline{x}_3$	x_1	$\overline{x}_3$	x_2	$\overline{x}_2$	136
$F_6 = x_1 x_2 + x_3$	x_1	x_2	x_1	x_3	x_1	x_1	x_1	$\overline{x}_1$	16
$F_7 = x_1 x_2 + x_2 x_3 + x_1 x_3$	x_1	x_2	x_1	x_3	x_1	$\overline{x}_3$	x_2	$\overline{x}_2$	11
$F_8 = x_1 x_2 + \overline{x}_1 x_3$	x_1	x_2	x_1	$\overline{x}_1$	x_1	x_1	x_3	$\overline{x}_3$	44
$F_9 = \overline{x}_1 x_2 x_3 + x_1 \overline{x}_2 + \overline{x}_3$	x_1	$\overline{x}_2$	x_1	$\overline{x}_3$	x_1	x_1	x_3	$\overline{x}_2$	337
$F_{10} = \overline{x}_1 \overline{x}_2 \overline{x}_3 + \overline{x}_1 x_2 x_3 + x_1 \overline{x}_2 x_3 + x_1 x_2 \overline{x}_3$	Pas de solution								1119
$F_{11} = x_1 + x_2 x_3$	x_1	x_1	x_1	x_2	x_1	$\overline{x}_3$	x_2	$\overline{x}_2$	10
$F_{12} = x_1 x_2 + x_1 x_3 + \overline{x}_2 x_3 + x_2 \overline{x}_3$	x_1	x_2	$\overline{x}_2$	x_3	$\overline{x}_2$	x_3	x_2	x_1	331
$F_{13} = \overline{x}_1 \overline{x}_2 \overline{x}_3 + \overline{x}_1 \overline{x}_2 x_3 + \overline{x}_1 x_2 \overline{x}_3 + x_1 \overline{x}_2 \overline{x}_3 + x_1 x_2 x_3$	$\overline{x}_1$	$\overline{x}_2$	$\overline{x}_3$	$\overline{x}_1$	x_2	$\overline{x}_2$	x_3	$\overline{x}_1$	953
$F_{14} = x_1 + x_2 \overline{x}_3 = \overline{x}_2 x_3$	x_1	x_1	x_2	$\overline{x}_3$	$\overline{x}_2$	x_2	x_3	x_1	567

Tableau VIII

CONCLUSION

L'objectif principal de cette étude a été une évaluation des propriétés logiques de certains réseaux cellulaires combinatoires et le développement de procédures de synthèse pour la réalisation systématique et efficace de tels réseaux.

Les problèmes principaux posés par l'étude des réseaux cellulaires se situent à quatre niveaux :

- choix d'une structure d'interconnexion.
- choix d'une cellule de base de complexité modérée.
- élaboration de méthodes de synthèse systématique.
- minimisation des dimensions du réseau et du nombre de connexions externes.

Les deux premiers niveaux sont en fait très imbriqués En effet, l'augmentation de la complexité de la cellule par exemple par l'adjonction d'éléments mémoires dans cette cellule se traduit inéluctablement par la nécessité d'une structure d'interconnexion plus élaborée. Nous nous sommes quant à nous, limités à l'étude de réseaux possèdant une structure d'interconnexion très simple et d'opérateurs logiques peu complexes. Un premier développement utile de ce mémoire consisterait en une sorte de classification des réseaux cellulaires à fonction de la complexité de leur structure d'interconnexion et des cellules de base. C'est un problème difficile surtout au niveau de la définition des cellules de base car nous ne savons pas à priori quels types d'opérateurs sont logiquement les plus efficaces pour l'obtention d'un comportement fonctionnel arbitraire dans les réseaux cellulaires combinatoires.

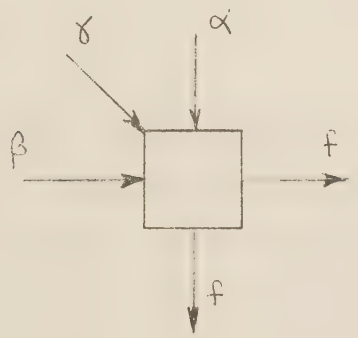
Au niveau de l'élaboration de méthodes de synthèse systématique, notre contribution a un double aspect.

Nous nous sommes efforcés dans un premier temps d'adapter des méthodes booléennes classiques au problème de la synthèse d'une fonction logique arbitraire par une structure régulière. C'est ainsi que nous avons introduit la notion de différence booléenne pour l'étude des cascades du type Maitra. Nous avons ainsi pu retrouver de façon simple des résultats déjà

connus par des approches complexes sur le nombre de fonctions réalisables par une cascade.

Nous avons développé une méthode de synthèse des réseaux en cascades qui se prête aisément à la programmation.

Toujours dans le cadre de l'application des méthodes booléennes classiques à l'étude des réseaux, nous avons élaboré une méthode de synthèse des fonctions logiques à l'aide de structures cellulaires composées d'opérateurs du type décrit sur la figure 1.



$$f = \gamma\overline{\beta} + \overline{\gamma}\,\alpha$$

Figure 1

Cette méthode est basée sur la décomposition de Shannon d'une fonction logique et conduit sous certaines réserves à la mise en place de réseaux cellulaires universels pour les fonction de n variables, et présentent un nombre minimum d'entrées externes [34]. Il serait certainement intéressant de développer des méthodes similaires pour d'autres types d'opérateurs dans la mesure où nous disposerions d'un critère permettant de juger de l'efficacité comparée de tel ou tel opérateur logique.

Nos efforts se sont ensuite portés sur la recherche de méthodes de synthèse spécifiques aux structures cellulaires. Nous avons été aidé dans ce domaine par les recherches menées dans l'équipe "Structures Logiques Nouvelles" du L.A.A.S. sur les propriétés des éléments à seuil. Ces propriétés nous ont permis d'aboutir à une caractérisation simple des réseaux à l'aide de systèmes d'inégalités linéaires en variables bivalentes. Cette

caractérisation est générale en ce sens qu'elle permet de traiter tout type d'opérateurs de base. Nous avons montré que la résolution de ce système d'inégalités linéaires était un problème de programmation linéaire pour lequel nous avons développé avec l'aide du service S. I. S. deux algorithmes inspirés de recherches poursuivies dans ce domaine aux Etats Unis.

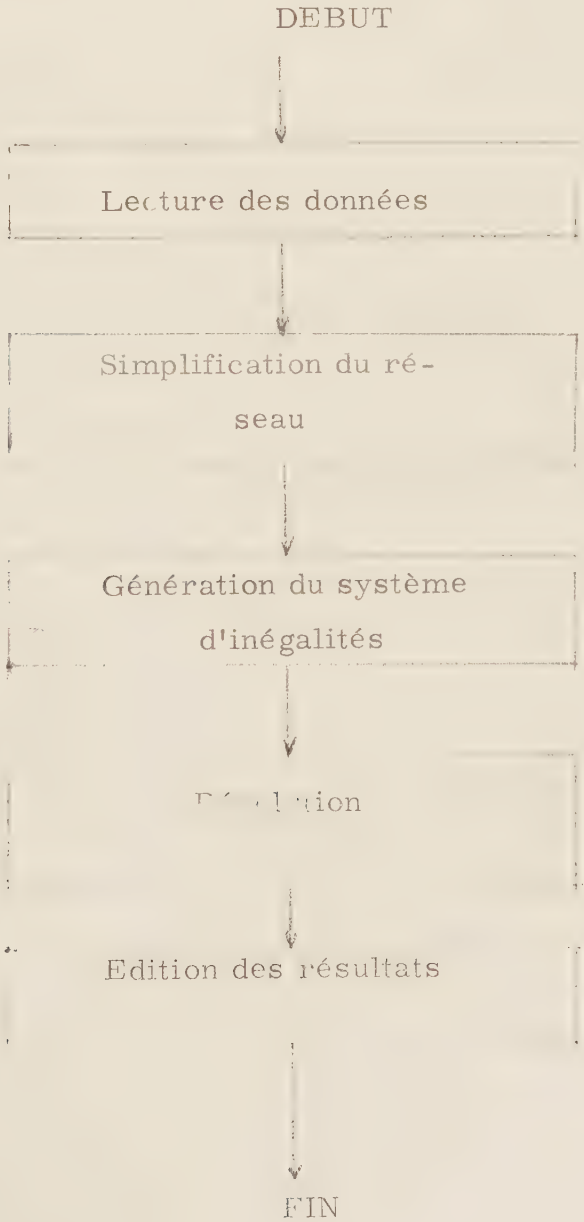
Le fait que ces algorithmes soient utiles en plus dans la résolution de nombreux problèmes logiques (couverture minimale, recherche de variables secondaires, ...) représente une retombée interessante de notre travail et nécessiterait un développement ultérieur, ne serait-ce qu' au niveau de l'amélioration du temps de calcul en fonction du nombre de variables.

Enfin, nous pensons que nos études doivent déboucher à plus ou moins brève échéance sur une classe plus étendue de réseaux à comportement séquentiel.

ANNEXE I

PREMIER ALGORITHME DE PROGRAMMATION LINEAIRE EN VARIABLES BIVALENTES

Dans cette annexe nous donnons le listing du programme de programmation linéaire en variables bivalentes étudié dans le chapitre III. Ce programme comprend plusieurs parties qui sont définies sur l'organigramme I.



Organigramme I

```
SN       SOURCE STATEMENT

 0 FIBFTC GLOP    NODECK
   C         PROGRAMME DE RESOLUTION SANS OPTIMISATION
   C         CAS D UN RESEAU 2 SUR 2  /  8 FONCTIONS DE 2 VARIABLES
   C         AUCUNE PRIORITE N EST ACCORDEE AUX DIFFERENTES VARIABLES
   C         DIMENSIONS CONSTANTES
 1         INTEGER P2
 2         INTEGER NOM(12)
 3         INTEGER CONT
 4         INTEGER U
 5         INTEGER P
 6         INTEGER FK
 7         INTEGER Y,W
10         INTEGER FMT1(3)
11         INTEGER FMT2(3)
12         INTEGER FMTL(5)
13         LOGICAL LJEX
14         LOGICAL API,NOSOLN
15         DIMENSION NUM (20)
16         DATA FMT1/5H(1HO,,1HA,4HI6/)/
17         DATA FMT2/5H(1HO,,1HA,5HI2//)/
20         DATA FMTL/1H(,1HA,1HI,1HA,1H)/
21         DATA IBLANC/1H /,IBARRE/1H//
22       1 FORMAT(3I3)
23       4 FORMAT(I3)
24       5 FORMAT(12A6)
25       6 FORMAT(1HO,15HTEMPS DE CALCUL,5X,I3,1HH,5X,I3,1HM,5X,I3,1HS,5X,I3,
          1     1HT//)
26       9 FORMAT(1H1,18HSEQUENCES CHOISIES////)
27    1030 FORMAT(1H1,50X,20HSYSTEME D INEGALITES///)
30    1023 FORMAT(1HO,31HLE PROBLEME N A PAS DE SOLUTION////)
31    1041 FORMAT(L2)
32    1039 FORMAT(1HO,20X,17HSOLUTION COURANTE//)
33    1022 FORMAT(1HO,19HPARAMETRE DU MODULE,I3,2X,9HASSIGNE A,I3/)
34    1021 FORMAT(1HO,6HENTREE,I3,2X,23HDU RESEAU CONNECTEE A X,I2,A1/)
35    1025 FORMAT(1HO,50X,19HNOMBRE D ITERATIONS,I8///)
36    1020 FORMAT(1H1,12A6/////)
   C         DIMENSIONS RELATIVES A LA FONCTION A REALISER
37         INTEGER F(4)
   C         DIMENSIONS RELATIVES AU RESEAU UTILISE
40         DIMENSION MAT(4,2),IMAT(4,2)
41         DIMENSION IP1(4),IP2(4)
42         DIMENSION IS(4,4),IQ(2,4),IR(4)
43         DIMENSION NMAT(4,4,3)
   C         DIMENSIONS RELATIVES AU SYSTEME D INEGALITES
44         INTEGER S(24),X(24),B(25),A(25,25),T(24)
   C
   C         LECTURE DES DONNEES
   C
45         READ(5,1) M,N,P
51         DO 21 I=1,P
52       21 READ(5,1) I1,(MAT(I,J),J=1,2)
61         READ (5,4) NEX
63         DO 900 JEX=1,NEX
64         K1=2**N
65         LJEX=.TRUE.
```

```
 SN        SOURCE STATEMENT

 66        IF(JEX.EQ.1) LJEX=.FALSE.
 71        READ(5,5) (NOM(I),I=1,12)
 76        FMTL(2)=NBCHAR(K1)
 77        FMTL(4)=1
100        READ(5,FMTL) (F(K),K=1,K1)
     C
     C
     C         PROGRAMME DE SIMPLIFICATION
     C
     C
105        IF(LJEX) GO TO 901
110        DO 701 I=1,P
111        IP1(I)=0
112        IP2(I)=0
113        DO 701 J=1,2
114        IMAT(I,J)=0
115    701 CONTINUE
120        IF(P.LE.8) GO TO 920
123        P2=P/2+MOD(P,2)
124        GO TO 921
125    920 P2=P
126    921 DO 702 I=1,P2
127        IR(I)=0
130        IQ(1,I)=0
131        IQ(2,I)=0
132        DO 702 J=1,P2
133        IS(I,J)=0
134    702 CONTINUE
137        DO 703 I=1,P
140        DO 703 J=1,2
141        IF(MAT(I,J).LE.0) GO TO 703
144        K=MAT(I,J)
145        IF(IMAT(K,1).GT.0) GO TO 704
150        IMAT(K,1)=I
151        GO TO 703
152    704 IMAT(K,2)=I
153    703 CONTINUE
156        DO 705 I=1,P
157        IF(IMAT(I,2).EQ.0) IP1(I)=1
162    705 CONTINUE
     C
     C         RECHERCHE D UN ETAT DEPART
     C
164        IDS1=1
165    799 DO 706 I=1,P
166        IF(IP1(I).EQ.1.AND.IP2(I).EQ.0) GO TO 707
171    706 CONTINUE
173        GO TO 750
     C
     C         ETAT DEPART
     C
174    707 IDR=0
175    708 IDR=IDR+1
176        IR(IDR)=I
177        IP2(I)=1
```

```
ISN       SOURCE STATEMENT

200       IF(I.EQ.1) GO TO 709
203       J=MAT(I,1)
204       IF(IP2(J).EQ.1) GO TO 709
207       I=J
210       GO TO 708
      C
      C         CONSTRUCTION SEQUENCE PRIMAIRE ED
      C
211     709 IDS2=1
212         IDQ=1
213         DO 710 I=1,IDR
214         I1=IDR-I+1
215         J=IR(I1)
216         IF(I1.EQ.1) GO TO 711
221         J1=IR(I1-1)
222         IS(IDS1,IDS2)=J
223         IDS2=IDS2+1
224         IP2(J)=1
225         IF(IMAT(J,1).NE.J1) GO TO 712
230         IQ(1,IDQ)=J
231         IQ(2,IDQ)=IMAT(J,2)
232         IDQ=IDQ+1
233         GO TO 710
234     712 IQ(1,IDQ)=J
235         IQ(2,IDQ)=IMAT(J,1)
236         IDQ=IDQ+1
237     710 CONTINUE
241     711 J=IR(1)
242     715 IF(J.EQ.P) GO TO 713
245         J1=IMAT(J,1)
246         IF(IP2(J1).EQ.1) GO TO 713
251         IS(IDS1,IDS2)=J
252         IDS2=IDS2+1
253         IP2(J)=1
254         J=J1
255         GO TO 715
256     713 IS(IDS1,IDS2)=J
257         IP2(J)=1
260         IDS1=IDS1+1
261         IDS2=1
262         IF(IDQ.EQ.1) GO TO 799
      C
      C         CONSTRUCTION SEQUENCES SECONDAIRES ED
      C
265     714 IDQ=IDQ-1
266         J=IQ(1,IDQ)
267         DO 716 I=1,P2
270         IE=IDS1-1
271         IF(IS(IE,I).EQ.J) GO TO 717
274         IS(IDS1,I)=IS(IE,I)
275     716 CONTINUE
277     717 IS(IDS1,I)=J
300         K=IQ(2,IDQ)
301         IS(IDS1,I+1)=K
302         IP2(K)=1
```

```
ISN        SOURCE STATEMENT

303        IDS1=IDS1+1
304        IF(IDQ.NE.1) GO TO 714
307        GO TO 799
     C
     C        CAS / PAS D ETAT DE DEPART
     C
310    750 DO 718 I=1,P
311        IF(IP2(I).EQ.0) GO TO 719
314    718 CONTINUE
316        GO TO 699
317    719 IDS2=1
320        IDQ=1
321    720 IS(IDS1,IDS2)=I
322        IDS2=IDS2+1
323        IP2(I)=1
324        J1=IMAT(I,1)
325        J2=IMAT(I,2)
326        IF(IP2(J1).EQ.1.OR.IP2(J2).EQ.1) GO TO 721
331        IQ(1,IDQ)=I
332        IQ(2,IDQ)=J2
333        IDQ=IDQ+1
334        I=J1
335        GO TO 720
336    721 IDS1=IDS1+1
337        IF(IDQ.EQ.1) GO TO 799
342        IDQ=IDQ-1
343        IE=IDS1-1
344        DO 722 I=1,P2
345        K=IQ(1,IDQ)
346        IF(IS(IE,I).EQ.K) GO TO 723
351        IS(IDS1,I)=IS(IE,I)
352    722 CONTINUE
354    723 IS(IDS1,I)=K
355        IP2(K)=1
356        IS(IDS1,I+1)=IQ(2,IDQ)
357        K=IQ(2,IDQ)
360        IP2(K)=1
361        GO TO 721
362    699 WRITE(6,9)
363        DO 760 I=1,P2
364        DO 761 J=1,P2
365        IF(IS(I,J).EQ.0) GO TO 762
370    761 CONTINUE
372    762 IF(J.EQ.1) GO TO 763
375        JY=J-1
376        JZ=JY
377        FMT1(2)=NBCHAR(JY)
400        WRITE(6,FMT1) (IS (I,J),J=1,JZ)
405    760 CONTINUE
407    763 CONTINUE
     C
     C        RECONFIGURATION DU RESEAU
     C
410        DO 790 I=1,P
411    790 IP1(I)=0
```

```
ISN       SOURCE STATEMENT

413           DO 800 I=1,P2
414           DO 801 J=1,P2
415           IF(IS(I,J).EQ.0) GO TO 802
420       801 CONTINUE
422       802 IF(J.EQ.1) GO TO 803
425           K=IS(I,J-1)
426           IP1(K)=I
427       800 CONTINUE
431       803 I1=I-1
432           DO 810 I=1,I1
433           L=IS(I,1)
434           NMAT(I,1,1)=L
435           L1=MAT(L,1)
436           L2=MAT(L,2)
437           IF(L1.LT.0) GO TO 811
442           NMAT(I,1,2)=IP1(L1)+100
443           GO TO 812
444       811 NMAT(I,1,2)=L1
445       812 IF(L2.LT.0) GO TO 813
450           NMAT(I,1,3)=IP1(L2)+100
451           GO TO 814
452       813 NMAT(I,1,3)=L2
453       814 DO 820 J=2,P2
454           L=IS(I,J)
455           IF(L.EQ.0) GO TO 816
460           NMAT(I,J,1)=L
461           L1=MAT(L,1)
462           L2=MAT(L,2)
465           IF(L1.EQ.NMAT(I,J-1,1)) GO TO 815
466           IF(L1.LT.0) GO TO 816
471           NMAT(I,J,2)=IP1(L1)+100
472           GO TO 820
473       816 NMAT(I,J,2)=L1
474           GO TO 820
475       815 IF(L2.LT.0) GO TO 817
500           NMAT(I,J,2)=IP1(L2)+100
501           GO TO 820
502       817 NMAT(I,J,2)=L2
503       820 CONTINUE
505       810 CONTINUE
507           Y=I1
      C
      C
      C          PROGRAMME DE GENERATION
      C
      C
510       900 NL=2*M+Y*2**(N+1)+1
511           NC=2*M*N+(Y-1)*2**N+P+1
512           JD=2*M*N+1
513             =JD+(Y-1)*2**N
514              NC-1
515              24 I=1,NL
516            D 24 J=1;NC
517        24 A(I,J)=0
      C
```

```
ISN        SOURCE STATEMENT

     C         FONCTION ECONOMIQUE
     C
522          DO 31 J=1,P
523          J1=JE+J
524       31 A(1,J1)=1
     C
     C         INEGALITES ASSOCIEES AUX LAMBDAS
     C
526          DO 41 I=1,M
527          A(2*I,1)=-1
530          A(2*I+1,1)=1
531          JA=2*(I-1)*N+2
532          JB=2*I*N+1
533          DO 41 J=JA,JB
534          A(2*I,J)=1
535          A(2*I+1,J)=-1
536       41 CONTINUE
541          IG=2*I-1
     C
     C         INEGALITES ASSOCIEES AUX CELLULES
     C
542          DO 51 I=1,Y
543          DO 51 K=1,K1
544          I1=IG+2*(I-1)*K1+2*K-1
545          I2=I1+1
546          DO 52 L=1,P2
547          IF(NMAT(I,L,1).EQ.0) GO TO 53
552       52 CONTINUE
554       53 NE=L-1
555          W=(-1)**NE
556          DO 54 L=1,3
557          LL=NMAT(I,1,L)
560          IF(LL.LT.0) GO TO 55
563          IF(LL.GT.100) GO TO 56
566          JA=JE+LL
567          A(I1,JA)=W
570          A(I2,JA)=-W
571          GO TO 54
572       55 LL=-LL
573          CALL DECOD (K,X,N,NV)
574          DO 57 IL=1,N
575          IF(X(IL).EQ.0) GO TO 58
600          JK=IL
601          GO TO 59
602       58 JK=N+IL
603       59 JA=2*(LL-1)*N+JK+1
604          A(I1,JA)=W
605          A(I2,JA)=-W
607       57 CONTINUE
610          GO TO 54
611       56 LL=LL-100
612          JA=JD+(LL-2)*K1+K
613          A(I1,JA)=W
614          A(I2,JA)=-W
615       54 CONTINUE
```

```
ISN         SOURCE STATEMENT

617         IF(NE.EQ.1) GO TO 51
622         DO 61 L=2,NE
623         W=W*(-2)
624         JA=JE+NMAT(I,L,1)
625         A(I1,JA)=W
626         A(I2,JA)=-W
627         LL=NMAT(I,L,2)
630         IF(LL.LT.0) GO TO 62
633         LL=LL-100
634         JA=JD+(LL-2)*K1+K
635         A(I1,JA)=W
636         A(I2,JA)=-W
637         GO TO 61
640      62 LL=-LL
641         CALL DECOD (K,X,N,NV)
642         DO 63 IL=1,N
643         IF(X(IL).EQ.0) GO TO 64
646         JK=IL
647         GO TO 65
650      64 JK=N+IL
651      65 JA=2*(LL-1)*N+JK+1
652         A(I1,JA)=W
653         A(I2,JA)=-W
654      63 CONTINUE
656      61 CONTINUE
660         W=0
661         DO 66 L=1,NE
662      66 W=W+(-1)**(NE+L+1)*2**L
664         W=W+1+(-1)**NE
665         W=-(W/2)
666         U=2**NE
667         IF(I.EQ.1) GO TO 81
672         JA=JD+(I-2)*K1+K
673         A(I1,JA)=-U
674         A(I2,JA)=U
675         A(I1,1)=U+W
676         A(I2,1)=-(W+1)
677         GO TO 51
700      81 FK=F(K)
701         IF(FK.EQ.1) GO TO 82
704         A(I1,1)=U+W
705         A(I2,1)=-(W+1)
706         GO TO 51
707      82 A(I1,1)=W
710         A(I2,1)=U-W-1
711      51 CONTINUE
714         WRITE(6,1030)
715         KA=0
716     311 K1=KA+1
717         KA=NC-KA
720         KA=MINI(KA,20)+K1-1
721         DO 311 J=K1,KA
722         J1=J-K1+1
723         NUM(J1)=J
724     311 CONTINUE
```

```
      SOURCE STATEMENT

      KO=KA-K1+1
      FMTL(2)=NBCHAR(KO)
      WRITE(6,FMTL) (NUM(J),J=1,KO)
      DO 312 I=1,NL
      WRITE(6,FMT1) (A(I,J),J=K1,KA)
  312 CONTINUE
      IF(NC.GT.KA) GO TO 310
C
C
C         PROGRAMME DE RESOLUTION
C
C
      NI=NL-1
      NV=NC-1
      READ(5,1041) API
C
C         INITIALISATION DE S
C
      IF(.NOT.API) GO TO 1100
      READ(5,4) NS
      KA=0
 1101 K1=KA+1
      KA=NS-KA
      KA=MINI(KA,16)+K1-1
      KO=KA-K1+1
      FMTL(2)=NBCHAR(KO)
      FMTL(4)=5
      READ(5,FMTL) (S(K),K=K1,KA)
      IF(KA.LT.NS) GO TO 1101
      GO TO 1102
 1100 DO 1103 I=1,NV
      S(I)=0
 1103 CONTINUE
      NS=0
 1102 CONTINUE
      CONT=0
      CALL [illegible]
C
C         TEST DE LA VALEUR DES ECARTS
C
 1110 CONTINUE
      CONT=CONT+1
      DO 1112 J=1,NV
      X(J)=0
 1112 CONTINUE
      [illegible] GO TO [illegible]
 1111 DO 1114 K=1,NS
      L=S(K)
      L1=MOD(L,4)
      L2=L/4
      X(L2)=MOD(L1,2)
 1114 CONTINUE
 1113 DO 1115 I=2,NL
      B(I)=A(I,1)
      DO 1115 J=1,NV
```

```
 ISN        SOURCE STATEMENT

 0033       B(I)=B(I)+A(I,J+1)*X(J)
 0034  1115 CONTINUE
 0037       DO1116 I=2,NL
 0040       IF(B(I).LT.0) GO TO 1117
 0043  1116 CONTINUE
 0045       NOSOLN=.FALSE.
 0046       GO TO 1150
 0047  1117 CONTINUE
      C
      C         RECHERCHE DES VARIABLES CANDIDATES
      C
 0050       DO 1120 I=1,NV
 0051       T(I)=1
 0052  1120 CONTINUE
 0054       IF(NS.EQ.0) GO TO 1125
 0057       DO 1121 I=1,NS
 0060       L=S(I)
 0061       L=L/4
 0062       T(L)=0
 0063  1121 CONTINUE
 0065  1125 CONTINUE
 0066       DO 1122 J=1,NV
 0067       IF(T(J).EQ.0) GO TO 1122
 0072       DO 1123 I=2,NL
 0073       IF(B(I).GE.0.OR.A(I,J+1).LT.0) GO TO 1123
 0076       GO TO 1122
 0077  1123 CONTINUE
 1101       T(J)=0
 1102  1122 CONTINUE
      C
      C         TEST DE POSSIBILITE SUR LA SOLUTION PARTIELLE
      C
 1104       DO 1130 I=2,NL
 1105       IF(B(I).GE.0) GO TO 1130
 1110       ISOM=0
 1111       DO 1131 J=1,NV
 1112       IF(T(J).EQ.0) GO TO 1131
 1115       IA=A(I,J+1)
 1116       ISOM=ISOM+MAXI(IA,0)
 1117  1131 CONTINUE
 1121       ISOM=B(I)+ISOM
 1122       IF(ISOM.LT.0) GO TO 1160
 1125  1130 CONTINUE
      C
      C         NOUVELLE SOLUTION PARTIELLE
      C
 1127       IDS=0
 1130       DO 1140 J=1,NV
 1131       IF(T(J).EQ.0) GO TO 1140
 1134       IDS=IDS+1
 1135       IF(IDS.EQ.1) JMAX=J
 1140       ISOM=0
 1141       DO 1141 I=2,NL
 1142       K=B(I)+A(I,J+1)
 1143       ISOM=ISOM+MINI(K,0)
```

```
 ISN        SOURCE STATEMENT

 144   1141 CONTINUE
 146        IF(IDS.EQ.1)  ISMAX=ISOM
 151        IF(ISOM.LE.ISMAX) GO TO 1140
 154        ISMAX=ISOM
 155        JMAX=J
 156   1140 CONTINUE
 160        NS=NS+1
 161        S(NS)=4*JMAX+1
 162        GO TO 1110
      C
      C          RECHERCHE DU DERNIER ELEMENT DE S NON SOULIGNE
      C
 163   1150 CONTINUE
 164        IF(NS.EQ.0) GO TO 1165
 167        DO 1161 I=1,NS
 170        I1=NS-I+1
 171        L=S(I1)
 172        L=MOD(L,4)
 173        IF(L.LE.2) GO TO 1180
 176   1161 CONTINUE
 200   1165 CONTINUE
 201        NOSOLN=.TRUE.
 202        GO TO 1150
      C
      C          BACK TRACKING
      C
 203   1180 CONTINUE
 204        L=S(I1)
 205        L=MOD(L,4)
 206        IF(L.EQ.0) GO TO 1181
 211        S(I1)=S(I1)+1
 212        GO TO 1182
 213   1181 S(I1)=S(I1)+3
 214   1182 CONTINUE
 215        NS=I1
 216        NT=NS+1
 217        DO 1183 I=NT,NV
 220        S(I)=0
 221   1183 CONTINUE
 223        GO TO 1110
      C
      C          EDITION DES RESULTATS
      C
1224   1150 CONTINUE
1225        CALL GIRL2(IAA,KH,KM,KS,KT)
1226    151 WRITE(6,1020) (NOM(I),I=1,12)
1233        WRITE(6,1025) CONT
1234        WRITE(6,6) KH,KM,KS,KT
1235        IF(NOSOLN) GO TO 152
1240        DO 151 I=1,M
1241        I1=L*N-(I-1)
1242        DO 151 J=1,N
1243        I1=I1+1
1244        IF(S(I1).EQ.1) GO TO 154
1247   151  CONTINUE
```

```
 ISN        SOURCE STATEMENT

.251        I1 ''+N
.252        DO 155 J=1,N
.253        I2=I1+J
.254        IF(X(I2).EQ.1) GO TO 156
.257    155 CONTINUE
.261    154 WRITE(6,1021) I,J,IBLANC
.262        GO TO 151
.263    156 WRITE(6,1021) I,J,IBAR.E
.264    151 CONTINUE
.266        DO 17? I=1,P
.267        I1=2*M*N+(Y-1)*2**N+I
.27?        WRITE(6,1022) I,X(I1)
.27?    17. CONTINUE
.27?        GO TO 900
.274    152 WRITE(6,1023)
.27?    9?. CONTINUE
?7          ??
```

```
>N        SOURCE STATEMENT

 0  FIBFTC MINI     NODECK
 1         FUNCTION MINI (I,J)
 2         IF(I.GE.J) GO TO 10
 5         MINI=I
 6         GO TO 20
 7      10 MINI=J
10      20 RETURN
11         END
```

```
ISN        SOURCE STATEMENT

  0 FIBFTC MAXI     NODECK
  1        FUNCTION MAXI(I,J)
  2        IF(I.GE.J) GO TO 10
  5        MAXI=J
  6        GO TO 20
  7     10 MAXI=I
 10     20 RETURN
 11        END
```

```
ISN          SOURCE STATEMENT

  0  $IBFTC DECOD    NODECK
  1         SUBROUTINE DECOD (M,Y,K,L)
  2         INTEGER Y(L)
  3         N=M-1
  4         DO 10 I=1,K
  5         J=K-I+1
  6         Y(J)=MOD(N,2)
  7         N=(N-Y(J))/2
 10      10 CONTINUE
 12         RETURN
 13         END
```

```
ISN        SOURCE STATEMENT

  0  FIBFTC NBCHAR  NODECK
  1         FUNCTION NBCHAR(I)
  2         J=I
  3         J1=MOD(J,1[illegible])
  4         J=(J-J1)/1[illegible]
  5         J2=MOD(J,1[illegible])
  6         J=(J-J2)/1[illegible]
  7         J3=MOD(J,1[illegible])
 10         NBCHAR=4096*J3+64*J2+J1
 11         RETURN
 12         END
```

ANNEXE II

DEUXIEME ALGORITHME DE PROGRAMMATION LINEAIR EN VARIABLES BIVALENTES

Dans cette annexe nous donnons le listing du deuxième programme de programmation linéaire en variables bivalentes étudié dans le chapitre III. Ce programme est construit comme le précédent et ne diffère qu'au niveau de l'algorithme de résolution.

```
C      ****************************************************************
C          DIMENSIONS CONSTANTES
        IMPLICIT INTEGER (A-K,M-Z)
        DIMENSION NOM(12)
        LOGICAL LJEX
        DATA IBLANC/1H /,IBARRE/1H//
      1 FORMAT(3I3)
      4 FORMAT(I3)
      5 FORMAT(12A4)
   1020 FORMAT(1H1,12A4//////)
   1025 FORMAT(1H0,50X,19HNOMBRE D ITERATIONS,I5///)
   1021 FORMAT(1H0,6HENTREE,I3,2X,23HDU RESEAU CONNECTEE A Y,I2,/1//
   1022 FORMAT(1H0,19HPARAMETRE DU MODULE,I3,2X,9H[illegible]SSIONE ,I2//)
   1023 FORMAT(1H0,31HLE PROBLEME N A PAS DE SOLUTION/////)
C          DIMENSIONS RELATIVES AU RESEAU
        DIMENSION MAT(4,2),IMAT(4,2)
        DIMENSION IP1(4),IP2(4)
        DIMENSION IS(5,5),IQ(2,5),IP(5)
        DIMENSION MMAT(5,5,3)
      3 FORMAT(1H0,5I4)
C          DIMENSIONS RELATIVES A LA FONCTION
        INTEGER F(4),FK
      2 FORMAT(4I1)
C          DIMENSIONS RELATIVES AU SYSTEME D INEGALITES
        INTEGER A(24,24),B(24)
        INTEGER Y(24),U(24),L(24),D(24)
        INTEGER S(2,24),X(24),V(24)
        INTEGER E(2,24),T(24),TS(24)
C          LECTURE DES DONNEES
        READ(5,1) M,N,P
        DO 21 I=1,P
     21 READ(5,1) I1,(MAT(I,J),J=1,2)
        READ(5,4) NEX
        DO 900 JEX=1,NEX
        LJEX=.TRUE.
        IF(JEX.EQ.1) LJEX=.FALSE.
        K1=2**N
        READ(5,5) (NOM(I),I=1,12)
        READ(5,2) (F(K),K=1,K1)
C          MISE EN CASCADES DES CELLULES
        IF(LJEX) GO TO 901
        DO 701 I=1,P
        IP1(I)=0
        IP2(I)=0
        DO 701 J=1,2
        IMAT(I,J)=0
    701 CONTINUE
        IF(P.LE.8) GO TO 920
        P2=P/2+MOD(P,2)
        GO TO 921
    920 P2=P
```

```
0045          721 CONTINUE
0046              DO 702 I=1,P2
0047              IR(I)=0
0048              IQ(1,I)=0
0049              IQ(2,I)=0
0050              DO 702 J=1,P2
0051              IS(I,J)=0
0052          702 CONTINUE
0053              DO 703 I=1,P
0054              DO 703 J=1,2
0055              IF(MAT(I,J).LE.0) GO TO 703
0056              K=MAT(I,J)
0057              IF(IMAT(K,1).GT.0) GO TO 704
0058              IMAT(K,1)=I
0059              GO TO 703
0060          704 IMAT(K,2)=I
0061          703 CONTINUE
0062              DO 705 I=1,P
0063              IF(IMAT(I,2).EQ.0) IP1(I)=1
0064          705 CONTINUE
              C         RECHERCHE D'U. ETAT [illegible]
0065              IDS1=1
0066          799 DO 706 I=1,P
0067              IF(IP1(I).EQ.1.AND.IP2(I).EQ.0) GO TO 707
0068          706 CONTINUE
0069              GO TO 780
              C         ETAT DEPART
0070          707 IDP=0
0071          708 IDP=IDP+1
0072              IP(IDP)=I
0073              IP2(I)=1
0074              IF(IDP.EQ.1) GO TO 709
0075              J=IP(IDP-1)
0076              IF(IP2(J).EQ.1) GO TO 709
0077              I=J
0078              GO TO 708
              C         CONSTRUCTION SEQUENCE PRIMAIRE ED
0079          709 IDS2=1
0080              IDQ=1
0081              DO 710 I=1,IDP
0082              I1=IDP-I+1
0083              J=IP(I1)
0084              IF(I1.EQ.1) GO TO 711
0085              J1=IP(I1-1)
0086              IS(IDS1,IDS2)=J
0087              IDS2=IDS2+1
0088              IP2(J)=1
0089              IF(IMAT(J,1).NE.J1) GO TO 712
0090              IQ(1,IDQ)=J
0091              IQ(2,IDQ)=IMAT(J,2)
0092              IDQ=IDQ+1
0093              GO TO 710
0094          712 IQ(1,IDQ)=J
0095              IQ(2,IDQ)=IMAT(J,1)
0096              IDQ=IDQ+1
0097          710 CONTINUE
0098          711 J=IR(1)
0099          715 IF(J.EQ.P) GO TO 713
```

```
              J1=IMAT(J,1)
              IF(IP2(J1).EQ.1) GO TO 713
              IS(IDS1,IDS2)=J
              IDS2=IDS2+1
              IP2(J)=1
              J=J1
              GO TO 715
          713 IS(IDS1,IDS2)=J
              IP2(J)=1
              IDS1=IDS1+1
              IDS2=1
              IF(IDQ.EQ.1) GO TO 799
C                CONSTRUCTION SEQUENCES STOCKAIRES ET
          714 IDQ=IDQ-1
              J=IQ(1,IDQ)
              DO 716 I=1,P2
              IF=IDS1-1
              IF(IS(IF,I).EQ.J) GO TO 717
              IS(IDS1,I)=IS(IF,I)
          716 CONTINUE
          717 IS(IDS1,I)=J
              K=IQ(2,IDQ)
              IS(IDS1,I+1)=K
              IP2(K)=1
              IDS1=IDS1+1
              IF(IDQ.EQ.1) GO TO 714
              GO TO 715
C                CAS / PAS DIETAT DE DEPART
          750 DO 718 I=1,P
              IF(IP2(I).EQ.0) GO TO 719
          718 CONTINUE
              GO TO 699
          719 IDS2=1
              IDQ=1
          720 IS(IDS1,IDS2)=I
              IDS2=IDS2+1
              IP2(I)=1
              J1=IMAT(I,1)
              J2=IMAT(I,2)
              IF(IP2(J1).NE.1.OR.IP2(J2).EQ.1) GO TO 721
              IQ(1,IDQ)=I
              IQ(2,IDQ)=J2
              IDQ=IDQ+1
              I=J1
              GO TO 720
          721 IDS1=IDS1+1
              IF(IDQ.EQ.1) GO TO 719
              IDQ=IDQ-1
              IF=IDS1-1
              DO 722 I=1,P2
              K=IQ(1,IDQ)
              IF(IS(IF,I).EQ.K) GO TO 723
              IS(IDS1,I)=IS(IF,I)
          722 CONTINUE
          723 IS(IDS1,I)=K
              IP2(K)=1
              IS(IDS1,I+1)=IQ(2,IDQ)
              K=IQ(2,IDQ)
```

```
FORTRAN IV G LEVEL 19                    MAIN                    DATE = 71300
                  A(I2,JA)=-C
                  LL=NMAT(I,C,2)
                  IF(LL.LT.0) GO TO 62
                  LL=LL-100
                  JA=JC+(LL-2)*K1+K
                  A(I1,JA)=C
                  A(I2,JA)=-C
                  GO TO 61
               62 LL=-LL
                  DO 63 IL=1,N
                  JK=IL+N*(1-X(IL))
                  JA=2*(LL-1)*N+JK
                  A(I1,JA)=C
                  A(I2,JA)=-C
               63 CONTINUE
               61 CONTINUE
                  C=0
               69 DO 66 C=1,NE
               66 C=C+(-1)**(NE+C+1)*2**C
                  C=C+1+(-1)**NE
                  C=-C/2
                  K=2**NE
                  IF(I.EQ.1) GO TO 81
                  JA=JB+(I-2)*K1+K
                  A(I1,JA)=-UC
                  A(I2,JA)=UC
                  B(I1)=UC+C
                  B(I2)=-(C+1)
                  GO TO 51
               81 FK=F(K)
                  IF(FK.EQ.1) GO TO 82
                  B(I1)=UC+C
                  B(I2)=-(C+1)
                  GO TO 51
               82 B(I1)=C
                  B(I2)=UC-C-1
               51 CONTINUE
C                    RESOLUTION DU SYSTEME D INEGALITES
C                    SYNTHESE DE LA FONCTION A L AIDE D ELEMENTS CELLULAIRES
                  CONT=C
                  NS=0
C                    INITIALISATION DU PROGRAMME DE RESOLUTION
                  DO 4000 J=1,NC
                  S(1,J)=C
                  S(2,J)=C
                  V(J)=C
            4000  X(J)=C
                  DO 4001 I=1,NL
                  Y(I)=B(I)
                  U(I)=B(I)
                  L(I)=B(I)
                  N(I)=C
                  DO 4001 J=1,NC
                  IF(A(I,J)) 4002,4003,4004
            4002  L(I)=L(I)+A(I,J)
                  GO TO 4003
            4004  U(I)=U(I)+A(I,J)
            4003  IA=IABS(A(I,J))
```

```
0323                IF(IA.LE.1) GO TO 4001
0324                IF(D(I).EQ.0) GO TO 4005
0325                D(I)=-1
0326                GO TO 4001
0327           4005 D(I)=J
0328           4001 CONTINUE
0329                DO 4006 I=1,NL
0330                T(I)=0
0331                TS(I)=0
0332                IF(L(I).LT.0) T(I)=1
0333           4006 CONTINUE
                C          VERIFICATION CYCLICITY
0334           4009 I=0
0335                CONT=CONT+1
0336                COEF=0
0337           4026 I=I+1
0338                IF(I.GT.NL) I=1
0339                IF(T(I).EQ.0) GO TO 4010
0340                T(I)=0
0341                IF(J(I)) 4200,4011,4039
0342           4010 IF(D(I).EQ.-1) GO TO 4011
0343                GO TO 4026
0344           4011 IA=0
0345                DO 4012 J=1,NC
0346                IF(V(J).EQ.1) GO TO 4012
0347                IF(A(I,J).EQ.0) GO TO 4012
0348                IA=IABS(A(I,J))
0349                IF(IA.LE.B(I)) GO TO 4012
0350                INDEX=INDEX+1
0351                V(1,INDEX)=J
0352           4012 CONTINUE
0353                IF(INDEX.EQ.0) GO TO 4027
0354           4013 COEF(J)=COEF(J)+1
0355                IF(COEF(J).GT.NL) GO TO 4130
0356                GO TO 4030
0357           4027 DO 4021 V=1,INDEX
0358                J=V(1,V)
0359                NS=NS+1
0360                IF(A(I,J).GT.0) GO TO 4022
0361                S(1,NS)=-J
0362                S(2,NS)=2
0363                Y(J)=1
0364                X(J)=0
0365                DO 4023 IL=1,NL
0366                IF(A(IL,J)) 4024,4023,4025
0367           4024 L(IL)=L(IL)-A(IL,J)
0368                IF(L(IL).GE.0) T(IL)=0
0369                GO TO 4023
0370           4025 U(IL)=U(IL)-A(IL,J)
0371                IF(L(IL).LT.0) T(IL)=1
0372           4023 CONTINUE
0373                GO TO 4021
0374           4022 S(1,NS)=J
0375                S(2,NS)=2
0376                Y(J)=1
0377                X(J)=1
0378                DO 4026 IL=1,NL
0379                Y(IL)=Y(IL)+A(IL,J)
0380                IF(A(IL,J)) 4027,4026,4028
```

```
FORTRAN IV G LEVEL 19                    MAIN                    DATE = 71300
      4027 U(IL)=U(IL)+A(IL,J)
           IF(L(IL).LT.0) T(IL)=1
           GO TO 4026
      4028 L(IL)=L(IL)+A(IL,J)
           IF(L(IL).GE.0) T(IL)=0
      4026 CONTINUE
      4021 CONTINUE
           DO 4032 IL=1,NL
           IF(U(IL).LT.0) GO TO 4200
      4032 CONTINUE
           CODE=0
           GO TO 4010
C          COPIA LA NUEVELLE VARIABLE DANS LA SOLUTION
      4100 CONTINUE
           DO 4032 IL=1,NL
           IF(Y(IL).LT.0) GO TO 4034
      4033 CONTINUE
           GO TO 4150
      4034 CONTINUE
           DO 4035 I=1,NL
      4035 TS(I)=T(I)
           DO 4101 J=1,NC
           E(1,J)=0
      4101 E(2,J)=0
           INDEX=0
           DO 4102 J=1,NC
           IF(M(J).EQ.1) GO TO 4102
           INDEX=INDEX+1
           E(2,INDEX)=J
           DO 4103 I=1,NL
           I1=L(I)
           I2=L(I)+A(I,J)
           IF(I1.LT.0.AND.I2.GE.0) E(1,INDEX)=E(1,INDEX)+1
      4102 CONTINUE
      4103 CONTINUE
           IF(INDEX.EQ.0) GO TO 4200
           IF(INDEX.EQ.1) GO TO 4120
           I1=INDEX-1
           DO 4105 I=1,I1
           J2=I+1
           DO 4105 J=J2,INDEX
           IF(E(1,I).GE.E(1,J)) GO TO 4105
           IA=E(1,I)
           E(1,I)=E(1,J)
           E(1,J)=IA
           IA=E(2,I)
           E(2,I)=E(2,J)
           E(2,J)=IA
      4105 CONTINUE
           DO 4106 I=2,INDEX
           IF(E(1,I).NE.E(1,1)) GO TO 410
      4106 CONTINUE
      4107 I1=I-1
           IF(I1.EQ.1) GO TO 4120
           DO 4108 K=1,I1
           J=E(2,K)
           ISOM=0
           DO 4109 I=1,NL
```

```
                  I2=L(I)+A(I,J)
                  IF(A(I,J).GT.0.AND.I2.LT.0) ISOM=ISOM+A(I,
            4109  CONTINUE
                  IF(K.EQ.1) GO TO 4110
                  IF(ISOM.LE.ISMAX) GO TO 4108
            4110  ISMAX=ISOM
                  E(2,1)=J
            4108  CONTINUE
            4120  J=E(2,1)
                  NS=NS+1
                  S(1,NS)=J
                  V(J)=1
                  X(J)=1
                  DO 4130 I=1,ML
                  Y(I)=Y(I)+A(I,J)
                  IF(A(I,J)) 4131,4130,4132
            4131  L(I)=L(I)+A(I,J)
                  IF(L(I).LT.0) T(I)=1
                  GO TO 4130
            4132  L(I)=L(I)+A(I,J)
                  IF(L(I).GE.0) T(I)=0
            4130  CONTINUE
                  DO 4135 IL=1,ML
                  IF(Y(IL).LT.0) GO TO 4100
            4135  CONTINUE
                  GO TO 4150
C                   RETOUR SUR LA SOLUTION          BACK TRACKI
            4200  CONTINUE
                  DO 4201 I=1,NS
                  II=NS-I+1
                  IF(S(2,II).EQ.0) GO TO 4202
                  J=S(1,II)
                  IF(J.GT.0) GO TO 4203
                  J=-J
                  DO 4204 K=1,ML
                  IF(A(K,J)) 4205,4204,4206
            4205  L(K)=L(K)+A(K,J)
                  GO TO 4204
            4206  L(K)=L(K)+A(K,J)
            4204  CONTINUE
                  GO TO 4207
            4203  DO 4208 K=1,ML
                  Y(K)=Y(K)-A(K,J)
                  IF(A(K,J)) 4209,4208,4210
            4209  L(K)=L(K)-A(K,J)
                  GO TO 4208
            4210  L(K)=L(K)-A(K,J)
            4208  CONTINUE
            4207  S(1,II)=0
                  S(2,II)=0
                  X(J)=0
                  V(J)=0
            4201  CONTINUE
                  GO TO 4300
            4202  J=S(1,II)
                  J=IABS(J)
                  S(1,II)=-S(1,II)
                  S(2,II)=2
```

```
      Y(J)=1-Y(J)
      DO 4220 I=1,NL
      T(I)=TS(I)
      Y(I)=Y(I)+A(I,J)*(2*X(J)-1)
      U(I)=U(I)+A(I,J)*(2*X(J)-1)

      L(I)=L(I)+A(I,J)*(2*X(J)-1)
      IF(X(J).EQ.0) GO TO 4222
      IF(A(I,J).LT.0.AND.L(I).LT.0) T(I)=1
      IF(A(I,J).GT.0.AND.L(I).GE.0) T(I)=0
      GO TO 4220
 4222 IF(A(I,J).GT.0.AND.L(I).LT.0) T(I)=1
      IF(A(I,J).LT.0.AND.L(I).GE.0) T(I)=0
 4220 CONTINUE
      NS=II
      GO TO 4005
C         EDITION DES RESULTATS
 4150 CONTINUE
       WRITE(6,1020) (NOM(I),I=1,12)
       WRITE(6,1025) CONT
      DO 151 I=1,N
      I1=2**M*(I-1)
      DO 153 J=1,M
      I2=I1+J
      IF(X(I2).EQ.1) GO TO 154
  153 CONTINUE
      I1=I1+M
      DO 155 J=1,M
      I2=I1+J
      IF(X(I2).EQ.1) GO TO 156
  155 CONTINUE
  154  WRITE(6,1021) I,J,IBLANC
      GO TO 151
  156 WRITE(6,1021) I,J,INVERS
  151 CONTINUE
      DO 170 I=1,P
      I1=NC-P+I
      WRITE(6,1022) I,X(I1)
  170 CONTINUE
      GO TO 900
 4000 CONTINUE
      WRITE(6,1020) (NOM(I),I=1,12)
      WRITE(6,1025) CONT
      WRITE(6,1023)
  900 CONTINUE
      STOP
      END
```

```
      C        PROGRAMME  CODAGE  DECIMAL - BINAIRE
0001        SUBROUTINE DECOD (M,Y,K,L)
0002        INTEGER Y(L)
0003        N=M-1
0004        DO 10 I=1,K
0005        J=K-I+1
0006        Y(J)=MOD(N,2)
0007        N=(N-Y(J))/2
0008     10 CONTINUE
0009        RETURN
0010        END
```

BIBLIOGRAPHIE

[1] HENNIE, F.C. (1961) "Iterative Arrays of Logical Circuits" MIT Press, Cambridge, Massachussets.

[2] WAHLSTROM, S.E. (1967) "Programmable Logic Arrays. Cheaper by the millions". Electronic, 11 Décembre, p. 90-95.

[3] MINNICK, R.C. (1967) "A Survey of Microcellular Logic". J. Ass. Comput. Machinery 14, p. 203-241.

[4] ELSPAS, B., et TURNER, J.P. (1968) "Theory of Cellular Logic Networks and Machines" Stanford Research Institute. Report AD 66 9825.

[5] MINNICK,R.C., SHORT, R.A., GOLDBERG, J., STONE, H.S., GREEN,M.W., YOELI , M., KAUTZ, W.H. (1966) 'Cellular Arrays for Logic and Storage". Stanford Res. inst., Project 5087, Contract AF 19 (628) 4233 ; AFCRL.

[6] KAUTZ, W.H., PEASE, M.C., GREEN, M.W. (1970) "Cellular Logic in memory Arrays". Stanford Res. Inst., Project 5509, Contract Non - 4833 (00), Office of Naval Research Department of the Navy.

[7] MAITRA, K.K. (1962) "Cascaded Switching Networks of two input flexible cells". IRE Trans. Electron. Comput. EC-11, p. 136-143.

[8] SHORT, R.A. (1965) "Two rail Cellular Arrays". Dans AFIPS Conf. Proc. 27, 1, Washington D.C., Spartan, p. 355-369.

[9] MINNICK, R.C. (1964) "Cutpoint Cellular Logic" IEEE Trans. Electron. Comput. EC-13, p. 685-698.

[10] MINNICK, R.C. (1964) "Cellular linear input Logic" Stanford Res. Inst. Project 4641, Contract AF 19 (628)-498, AFCRL ou AD 433 802.

[11] ELSPAS, B., et SHORT, R.A. (1964) "A bound on the run measure of switching function". IEEE Trans. Electron. Comp. EC 13, 1.

[12] CALDWELL, S.H. (1958) "Switching Circuits and Logical Design". Wiley, New York.

[13] COHN, M. (1962) "Inconsistent cononical forms of switching functions". IRE Trans. Electron. Comput. EC-11, p. 284-285.

[14] KAUTZ, W.H. (1967) "A cellular threshold array". IEEE Trans. Comput. EC-16, p. 680-682.

[15] KAUTZ, W.H., LEVITT, K.N. et WAKSMAN, A. (1968) "Cellular Interconnection Arrays" IEEE Trans. Comput. C-17, p. 433-445.

[16] MUKHOPA DHYAY, A. (1969) "Unate Cellular Logic" IEEE Trans. Comput. C-18, p. 114-121.

[17] WEISS, C.D. (1969) "The Characterization and Properties of Cascade Realizable Switching Functions" IEEE Trans. Comput. C-18, pp. 624-633.

[18] BERTRAND, J.C., IPPOLITO, J.C., DURANTE, C., PRUNET, F. (1971) "Méthode de synthèse graphique des fonctions booléennes à l'aide d'opérateurs NI et ON". Automatisme, Mars 1971.

[19] BERTRAND, J.C., IPPOLITO, J.C. (1969) 'Conditions d'identité et d'interchangeabilité d'opérateurs logiques". CRAS, t. 269, pp. 168-171.

[20] BERTRAND, J.C., IPPOLITO, J.C., DURANTE, C. (1969) "Application des matrices booléennes à la synthèse des fonctions logiques à l'aide d'opérateurs NI et ON". Colloque Systèmes Logiques. Conceptions. Bruxelles.

[21] IPPOLITO, J.C., BERTRAND, J.C., DURANTE, C. (1969) 'Structures cellulaires utilisant des opérateurs NI-ON" Colloque Systèmes Logiques. Bruxelles.

[22] SHEN, V. (1969) "On Simple Disjonctive Decomposition of Switching Functions". Princeton University, Ph. D.

[23] AZEMA, P. (1970) "Contribution à la synthèse de fonctions logiques à l'aide d'éléments à seuil". Thèse de Doctorat ès Sciences Physiques, Université Paul Sabatier, Toulouse.

[24] LEWIS, P.M., COATES, C.L. (1967) "Threshold Logic". Wiley, New York.

[25] AZEMA P., IPPOLITO, J.C. (1970) "Etude de réseaux cellulaires composés d'opérateurs NI-ON à l'aide du système associée d'inégalités linéaires". Congrés d'Informatique de l'AFCET, Paris.

[26] MUROGA, S., IBARAKI, T. (1968) "Logical Design of an Optimal Network by Integer Linear Programming. Part I and Part II". Reports n° 264 et n° 269, Department of Computer Science, University of Illinois. July.

BALAS, E. (1965) "An additive algorithm for solving linear programs with zero one variables". Operations Research, 13, pp. 517-546.

BALINSKI, M.L. (1965) "Integer Programming : Methods, Uses, Computation". Management Sci. , 12, pp. 253-313.

GLOVER F. (1965) " A multiphase dual algorithm for the zero one integer programmin problem". Operations Research, 13, pp. 879-919.

GEOFFRION, A.M. (1967) "Integer programming by implicit enumeration and Balas Method" Siam Review, vol. 9, n° 2, pp. 178-190.

IBARAKI, I., LIU, T.K., BAUGH, C.R., MUROGA, S. (1969) "Implicit enumeration program post zero one integer programming", University of Illinois, Report 305.

AZEMA, P., IPPOLITO, J.C. (1971) "Une approche au test et à la synthèse de réseaux cellulaires composés de NI-ON". Note technique Interne LAAS n° 738.

IPPOLITO, J.C. (1971) "Implantation de fonctions logiques sur des réseaux cellulaires". Journée d'Etude AFCET, Paris, Décembre.

CANADAY, R.H. (1965) "Two dimensional iterative logic". Proceedings Fall Joint Computer Conference, pp. 343-353.

TABLE DES MATIERES

Introduction 1

Chapitre I - Description de quelques méthodes et techniques de la logique cellulaire 7

I-1 - Logique cellulaire et intégration à large échelle 9

I-1-1 - Evolution des techniques de fabrication de circuits intégrés : intégration à large échelle 9

I-1-2 - Intérêt de la logique cellulaire dans l'intégration à grande échelle 11

I-2 - Méthodes et techniques de la logique cellulaire 12

I-2-1 - Réseaux cellulaires non spécialisés 12

I-2-1-1 - Cascade de Maitra 12

I-2-1-2 - Cascades de Short. Cascade double rail 14

I-2-1-3 - Réseaux bidimensionnels 17

I-2-1-3-1 - Réseaux à points de coupure 18

I-2-1-3-2 - Réseaux additionneurs 20

I-2-1-3-3 - Réseaux à fonctions symétriques 23

I-2-1-3-4 - Réseaux cellulaires basés sur la décomposition d'une fonction logique sous forme de sommes de mintermes ou d'implicants premiers 24

I-2-1-3-5 - Réseaux cellulaires basés sur la décomposition galoisienne d'une fonction logique 30

I-2-1-3-6 - Réseaux de portes NI 32

I-2-2 - Réseaux spécialisés 34

I-2-2-1 - Réseaux cellulaires pour la réalisation de fonctions à seuil 34

I-2 2-2 - Réseaux cellulaires d'interconnexion 39

I-3 - Minimisation des réseaux cellulaires 44

I-4 - Conclusion 46

Chapitre II - Analyse et synthèse de circuits cellulaires particuliers à partir de méthodes booléennes classiques 47

II-1 - Analyse et synthèse de cascades de Maitra 49

II-1-1- Rappels sur les cascades de Maitra 49

II-1-2- Définition et propriétés d'une cellule de base composée d'opérateurs NI-ON 50

II-1-2-1 -Propriétés des opérateurs NI-ON 50

II-1-2-2- Réalisations électroniques d'un opérateur NI-ON 51

II-1-2-3- Cellule de base de la cascade 51

II-1-3- Méthodes d'énumération des fonctions réalisables par une structure en cascade 55

II-1-3-1 - Première méthode d'énumération 56

II-1-3-2- Deuxième méthode d'énumération 61

II-1-4- Tests permettant de déterminer si une fonction est réalisable par une structure en cascade 66

II-1-4-1- Définition des tests 66

II-1-4-2- Exemplesd'application 67

II-2- Synthèse de fonctions logiques à l'aide d'un réseau cellulaire bidimensionnel 68

II-2-1- Description de la structure choisie 69
II-2-2- Description de la cellule de base du réseau 70
II-2-3- Propriétés de la cellule de base 71
II-2-4- Etude d'un réseau de dimensions N=4, C=8. Synthèse d'une fonction de trois variables 72
II-2-5- Synthèse d'une fonction de n variables 76

II-2-5-1- Cas particulier n=4 76
II-2-5-2- Cas général (n quelconque) 81

II-3- Conclusion 84

Chapitre III - Etude de réseaux cellulaires à l'aide du système associé d'inégalités linéaires 85

Première partie - Formulation du problème 87

III-1-1- Rappels sur les fonctions à seuil 89

III-1-1-1- Définitions 89
III-1-1-2- Normalisation du système I 89
III-1-1-3- Différentes représentations d'une fonction à seuil 90
III-1-1-4- Propriétés des fonctions à seuil 93

III-1-2- Description d'un réseau cellulaire par un système d'inégalités linéaires 97

III-1-2-1- Système d'inégalités associé à la cellule de base 98
III-1-2-2- Passage d'un système de relations conditionnelles à un système d'inégalités linéaires 100
III-1-2-3- Système d'inégalités linéaires associé à un réseau cellulaire 102
III-1-2-4- Formulation du problème de la résolution du système d'inégalités linéaires associé à un réseau cellulaire en un problème de programmation linéaire 104

III-1-2-5-Méthodes et algorithmes de programmation linéaire en variables bivalentes 106

III-1-2-5-1-Principe de l'énumération implicite 109

III-1-2-5-2-Premier algorithme de programmation linéaire en variables bivalentes 111

III-1-2-5-3-Second algorithme de programmation linéaire en variables bivalentes 112

III-1-2-5-4-Conclusion 120

Deuxième partie - Exemples d'application 125

III-2-1-Etude de réseaux cellulaires composés d'opérateurs NI-ON 127

III-2-1-1-Définition de la cellule de base du réseau 127

III-2-1-2-Système d'inégalités associé à la cellule de base 128

III-2-1-3-Application à une structure régulière de 4 cellules de base 131

III-2-1-4-Dimensions du problème 134

III-2-1-5-Résultats expérimentaux 134

III-2-1-5-1-Réseau de 4 cellules 134

III-2-1-5-2-Réseau de 9 cellules 135

III-2-1-5-3-Conclusions 137

III-2-1-6-Réduction des dimensions du problème. Simplification du réseau 138

III-2-1-6-1-Théorème 138

III-2-1-6-2-Conséquences du théorème précédent 143

III-2-1-6-3-Résultats expérimentaux 145

III-2-2-Etude de réseaux cellulaires composés d'opérateurs majorité 149

III-2-2-2-1-Description de la cellule de base 149

III-2-2-2-2-Système d'inégalités associé à la cellule de base 149

III-2-2-2-3-Résultats expérimentaux 150

III-2-2-4-Propriétés particulières des réseaux d'opérateurs majorité 152

III-2-3-Réseaux cellulaires composés d'opérateurs particuliers 154

III-2-3-1-Définition de la cellule de base 154

III-2-3-2-Description du réseau par un système d'inégalités 154

III-2-3-3-Résultats expérimentaux 155

Conclusion 157

Annexe 1 163

Annexe 2 181

Bibliographie 195

Autorisation de soutenance
accordée.
TOULOUSE, le 16 NOV. 1971

Le Président
de l'Université PAUL-SABATIER
L. LARENG

UNIVERSITE PAUL SABATIER - TOULOUSE